Lecture 13: Optical flow

COS 429: Computer Vision

Princeton University

Slide deck credit: David Fouhey

https://www.youtube.com/watch?v=G3QrhdfLCO8

Optical Flow

Idea first introduced by psychologist JJ Gibson in ~1940s to describe how to perceive opportunities for motion

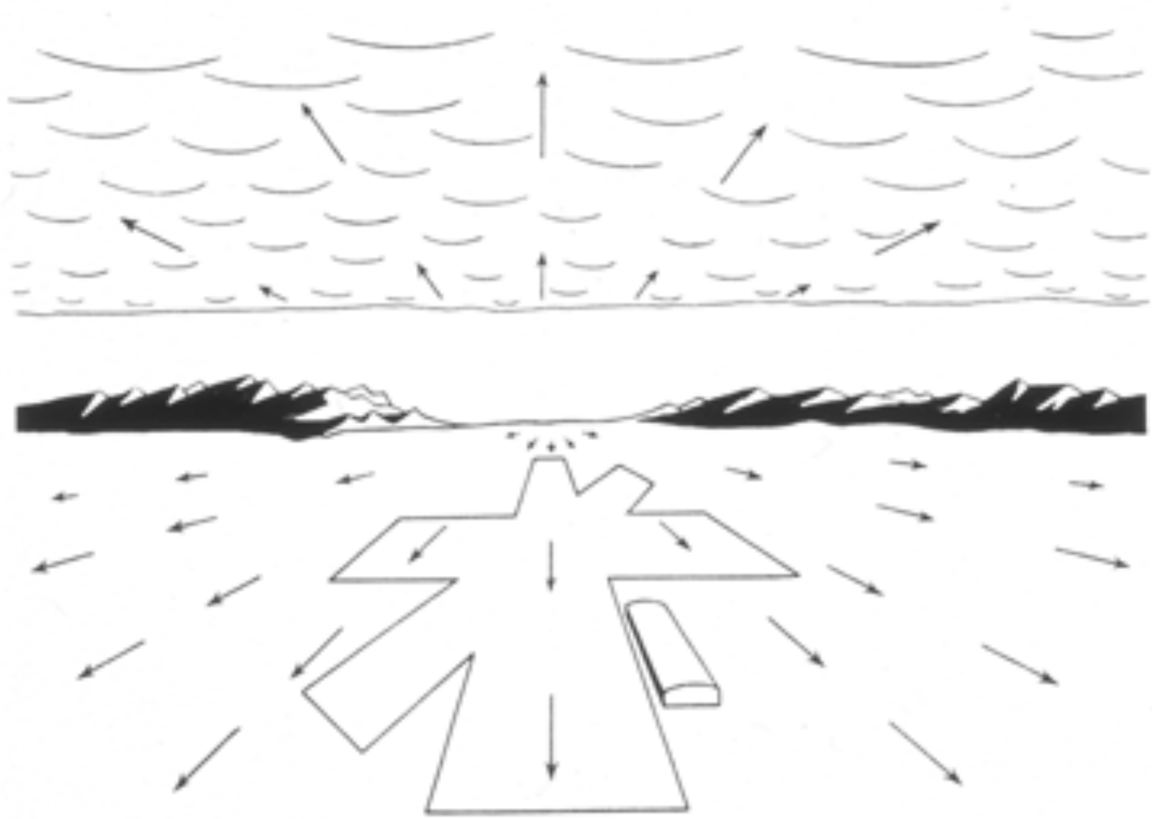
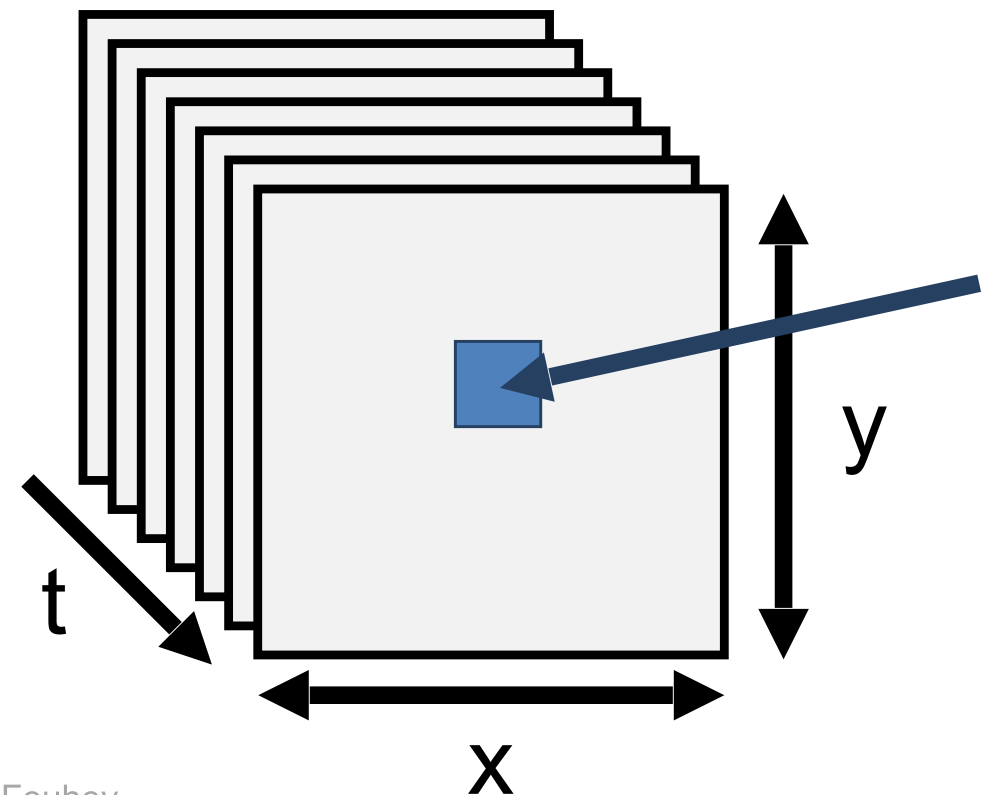


Image Credit: Gibson

Video

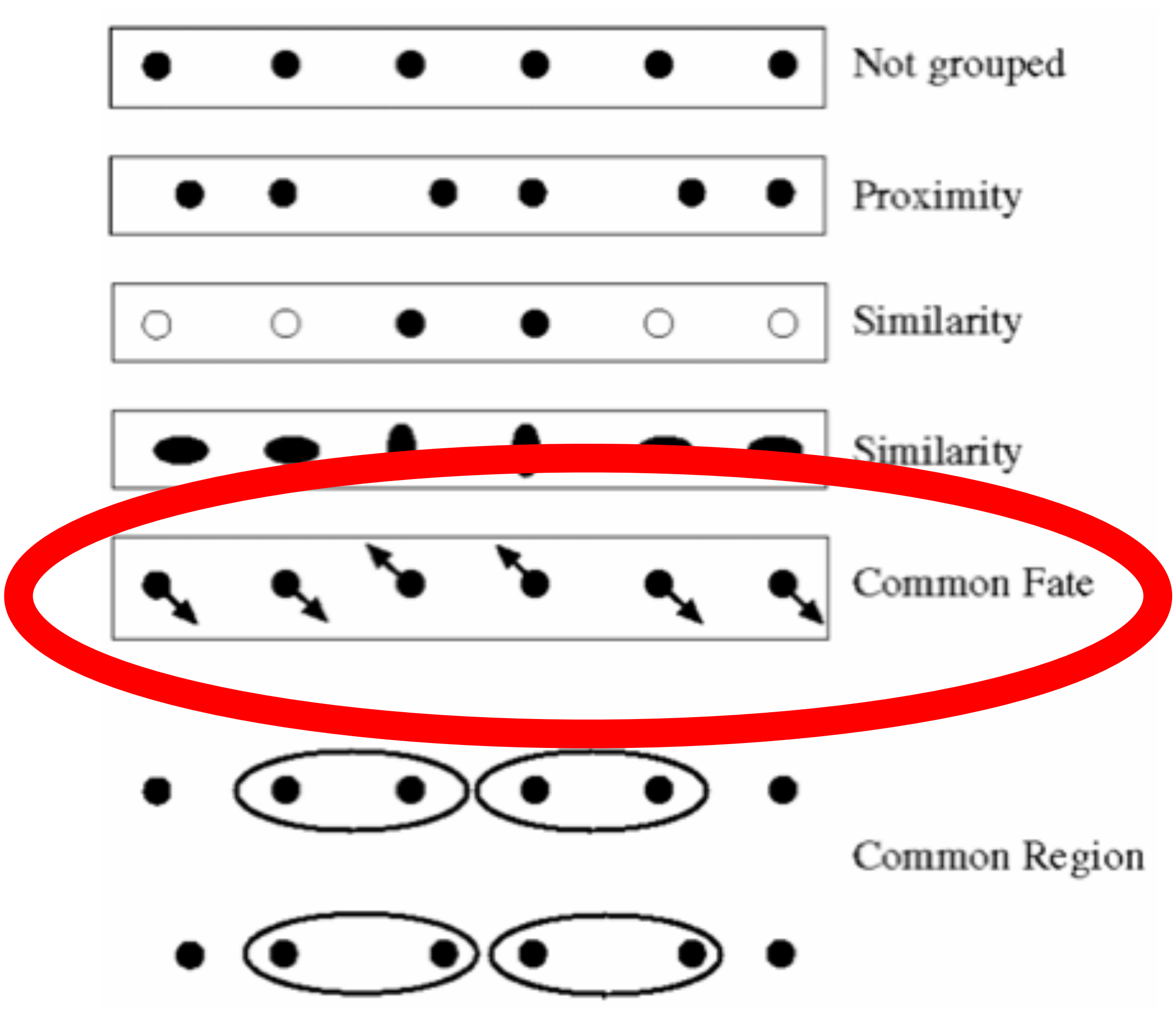
Video: sequence of frames over time
Image is function of space (x,y) and time t
(and channel c)



I(x,y,c,t)
x,y – location
c – channel
t – time

Credit: David Fouhey

Motion Perception

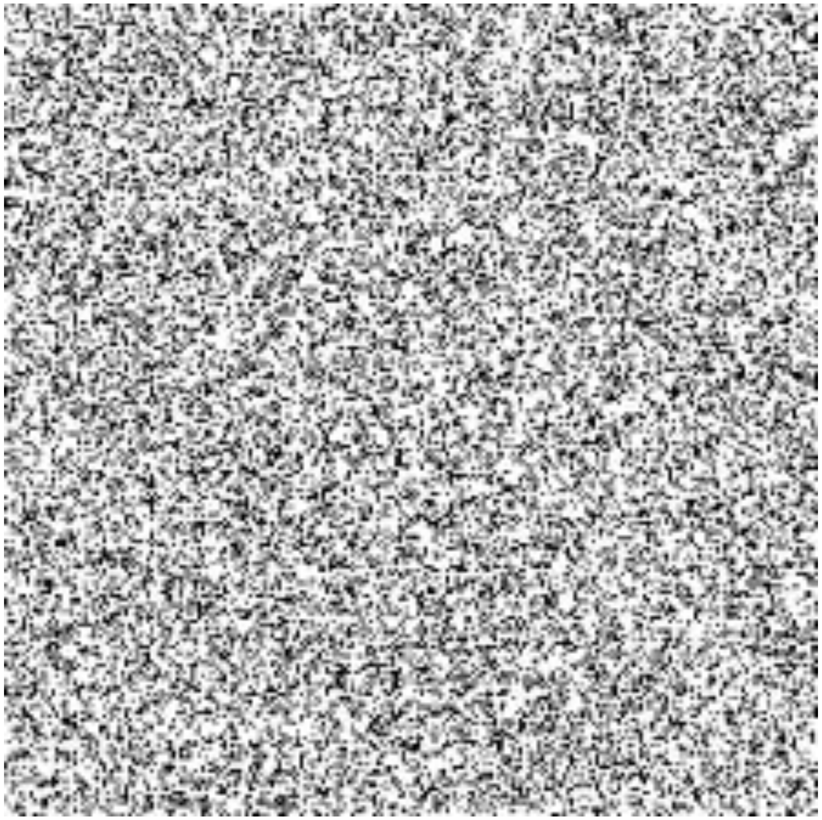


Gestalt psychology
Max Wertheimer
1880-1943

Slide Credit: S. Lazebnik

Motion and perceptual organization

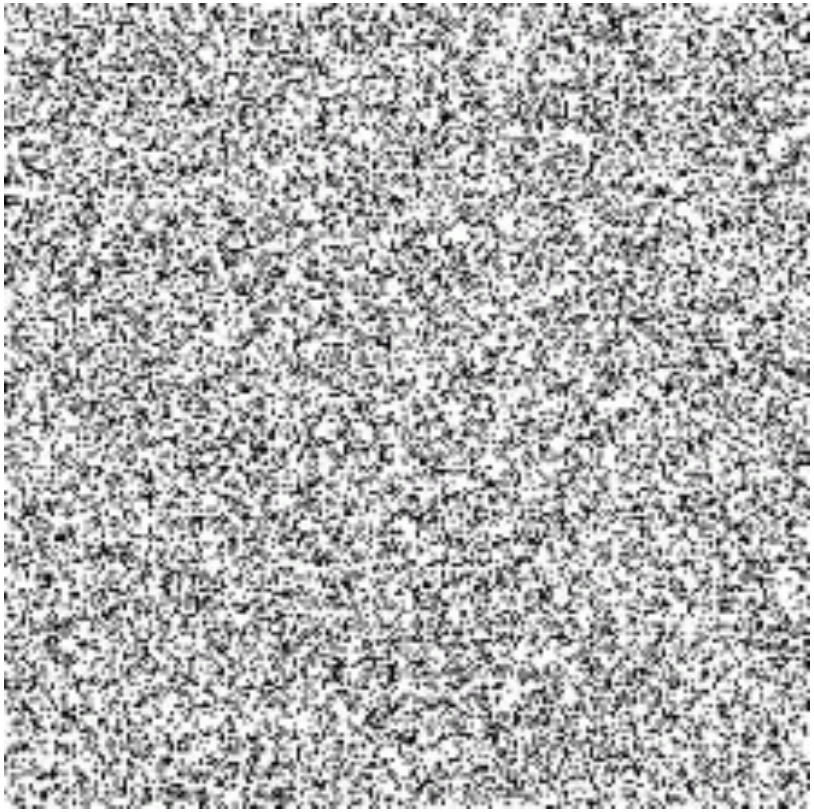
Sometimes motion is the only cue



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Motion and perceptual organization

Sometimes motion is the only cue



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Motion and perceptual organization

Even impoverished motion data can create a strong percept

Slide Credit: S. Lazebnik

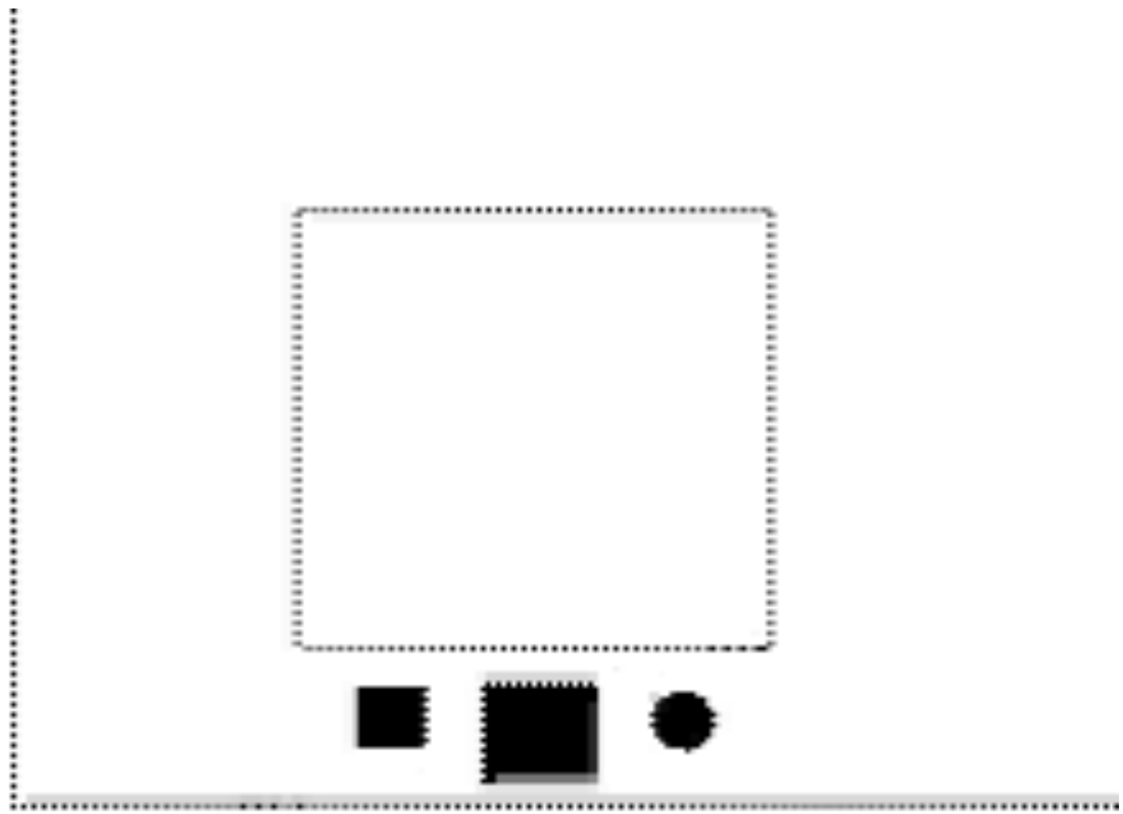
Motion and perceptual organization

Even impoverished motion data can create a strong percept

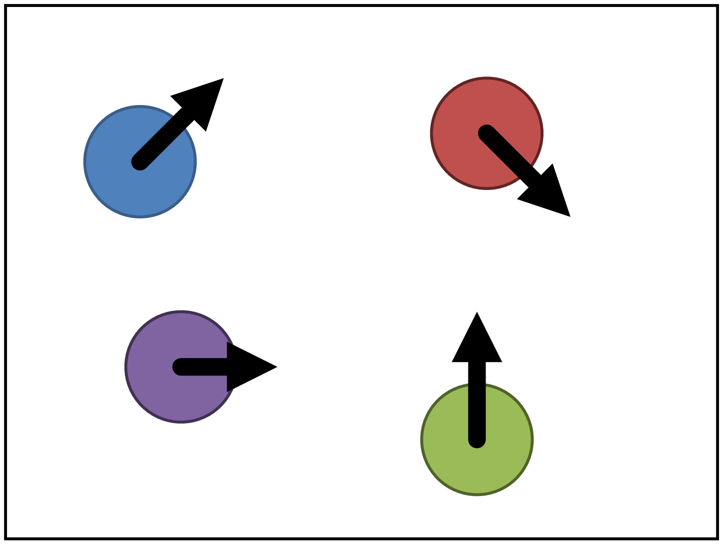
Slide Credit: S. Lazebnik

Motion and perceptual organization

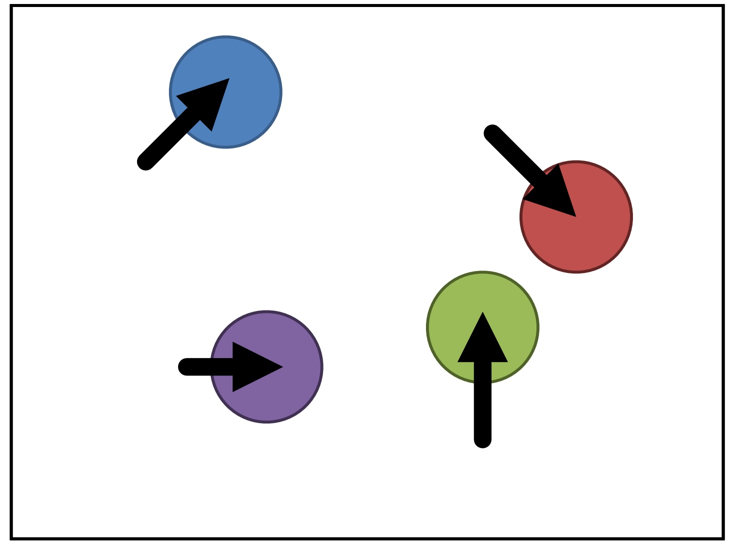
Even impoverished motion data can create a strong percept



Problem Definition: Optical Flow



I(x,y,t)



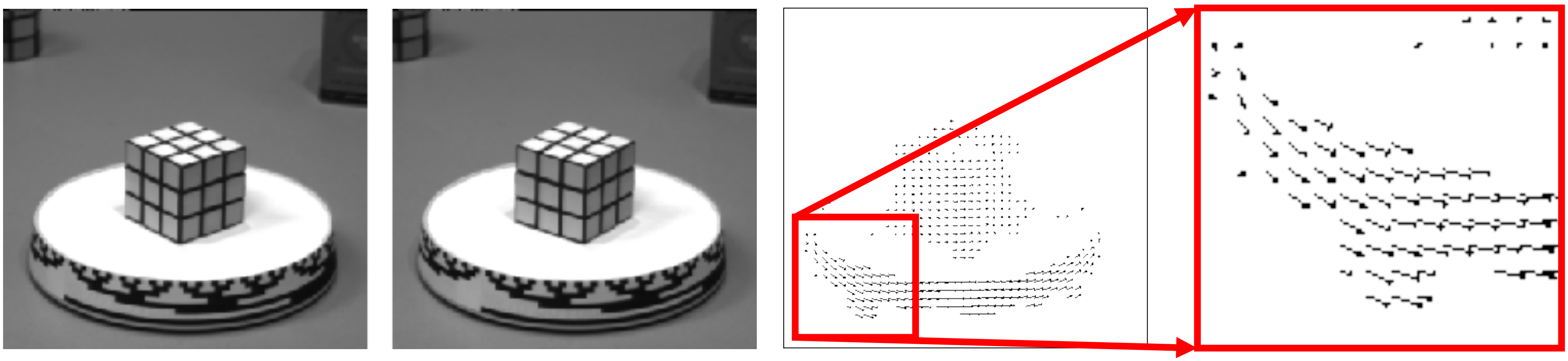
I(x,y,t+1)

Want to estimate pixel motion from image I(x,y,t) to image I(x,y,t+1)

Credit: David Fouhey

Optical flow

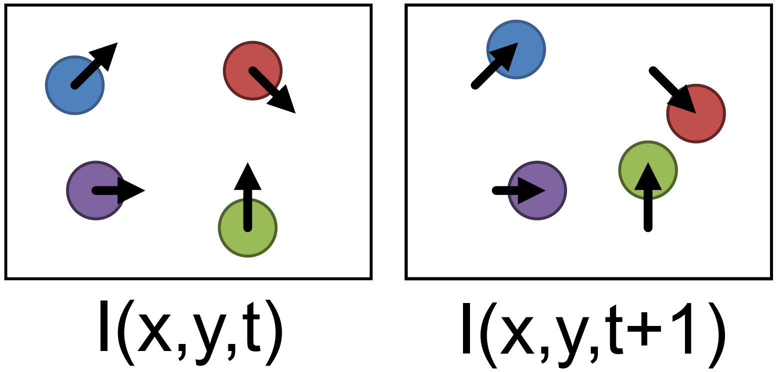
Optical flow is the *apparent* motion of objects



Will start by estimating motion of each pixel separately
Then will consider motion of entire image

Credit: David Fouhey

Optical Flow



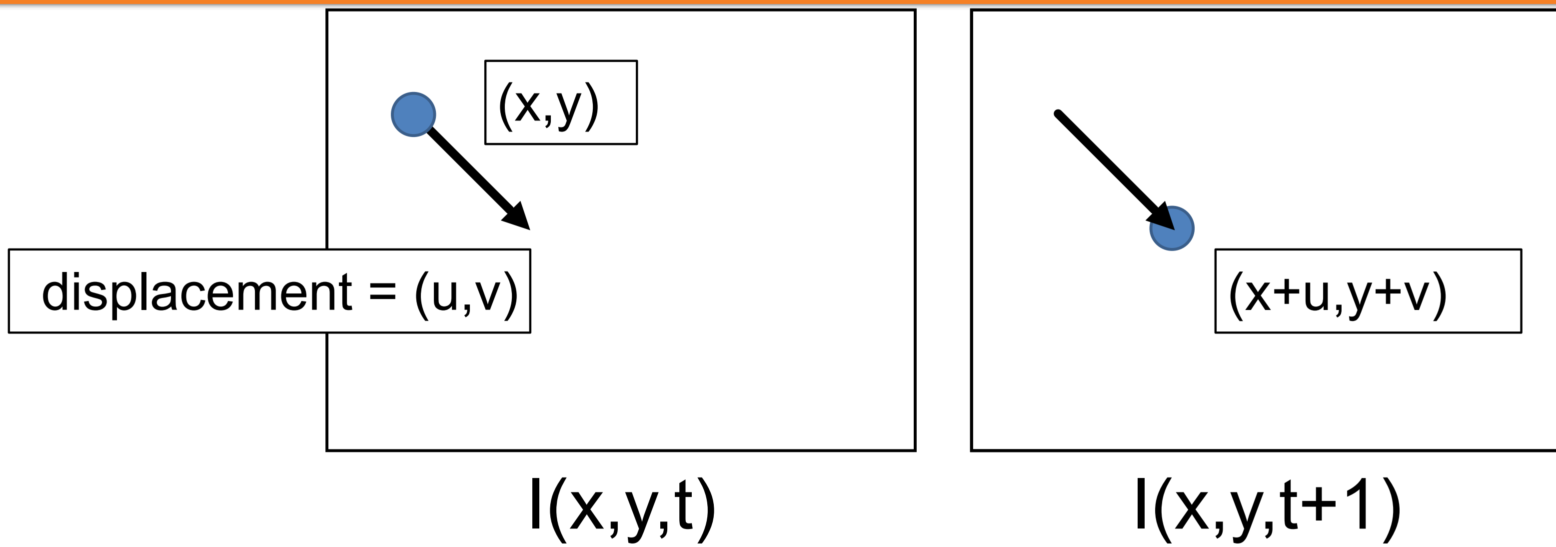
Solve correspondence problem: given pixel at time t, find **nearby** pixels of the **same color** at time t+1

Key assumptions:

- **Color/brightness constancy**: point at time t looks same at time t+1
- **Small motion**: points do not move very far

Credit: David Fouhey

Optical Flow

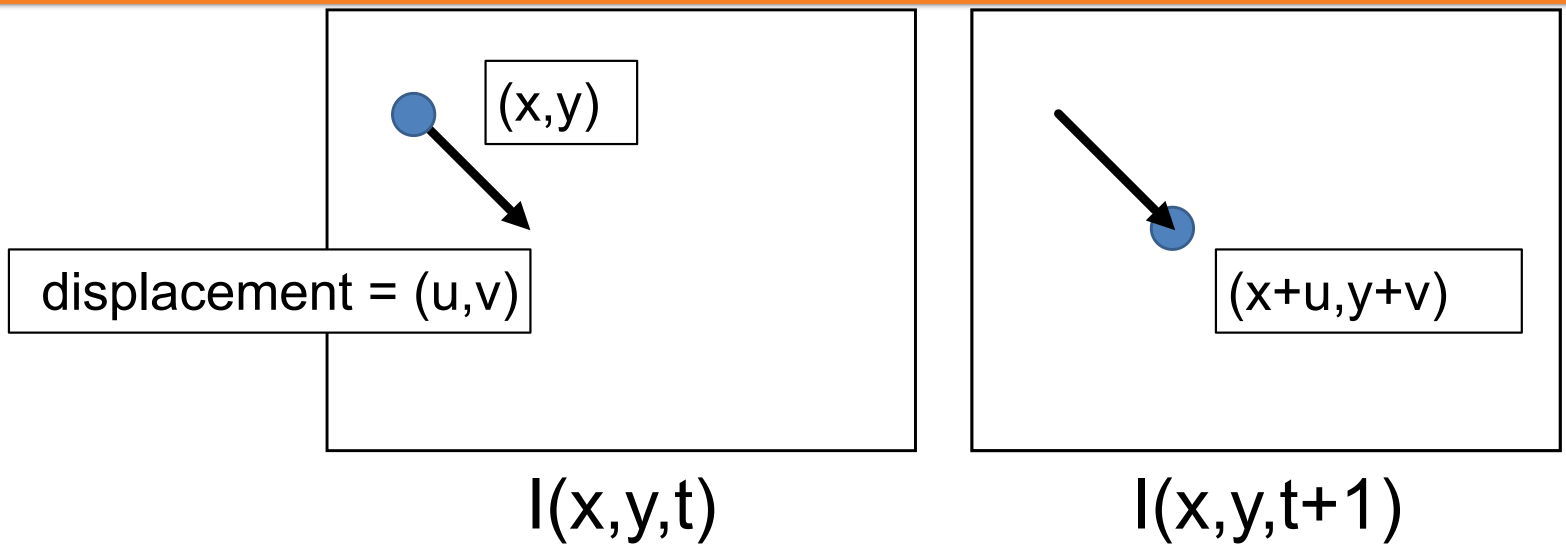


Brightness constancy: $$I(x, y, t) = I(x + u, y + v, t + 1)$$

Wrong way to do things: brute force match

Credit: David Fouhey

Optical Flow



Brightness constancy:

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

Recall Taylor Expansion:

$$I(x + u, y + v, t) = I(x, y, t) + I_x u + I_y v + \ldots$$

Credit: David Fouhey

Optical Flow Equation

$$I(x+u, y+v, t+1) = I(x, y, t)$$

$$\begin{aligned} 0 &= I(x+u, y+v, t+1) - I(x, y, t) \\ &\approx I(x, y, t+1) + I_x u + I_y v - I(x, y, t) \\ &= \underbrace{I(x, y, t+1) - I(x, y, t)} + I_x u + I_y v \end{aligned}$$

Taylor Expansion

If you had to guess, what would you call this?

Adapted from S. Lazebnik slides

Optical Flow Equation

$$I\left(x+u, y+v, t+1\right) = I(x, y, t)$$

$$\begin{aligned} 0 &= I\left(x+u, y+v, t+1\right) - I(x, y, t) \\ &\approx I\left(x, y, t+1\right) + I_x u + I_y v - I(x, y, t) \\ &= I\left(x, y, t+1\right) - I(x, y, t) + I_x u + I_y v \\ &= I_t + I_x u + I_y v \\ &= I_t + \nabla I \cdot [u, v] \end{aligned}$$

Taylor Expansion

When is this approximation exact?

[u,v] = [0,0]

When is it bad?

u or v big.

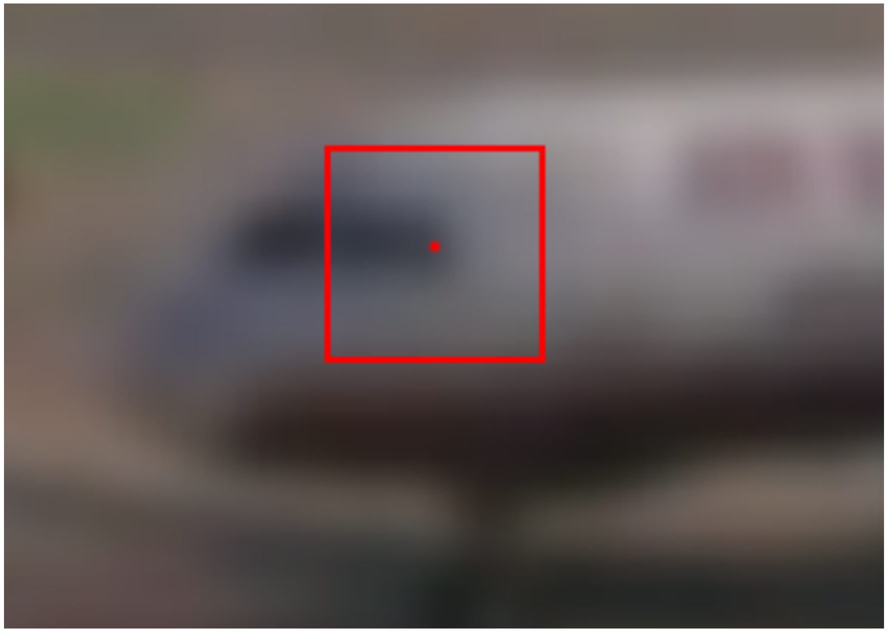
Adapted from S. Lazebnik slides

Optical Flow Equation

Brightness constancy equation

$$I_x u + I_y v + I_t = 0$$

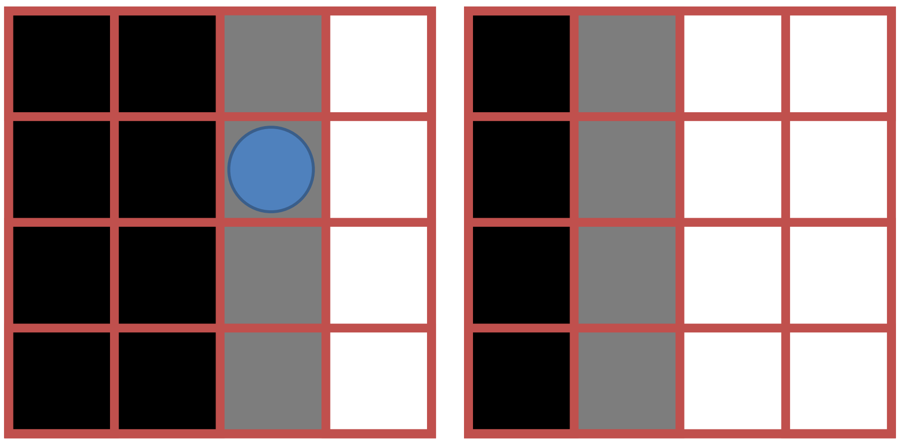
What do static image gradients have to do with motion estimation?



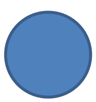
Slide Credit: S. Lazebnik

Brightness Constancy Example

$$I_x u + I_y v + I_t = 0$$

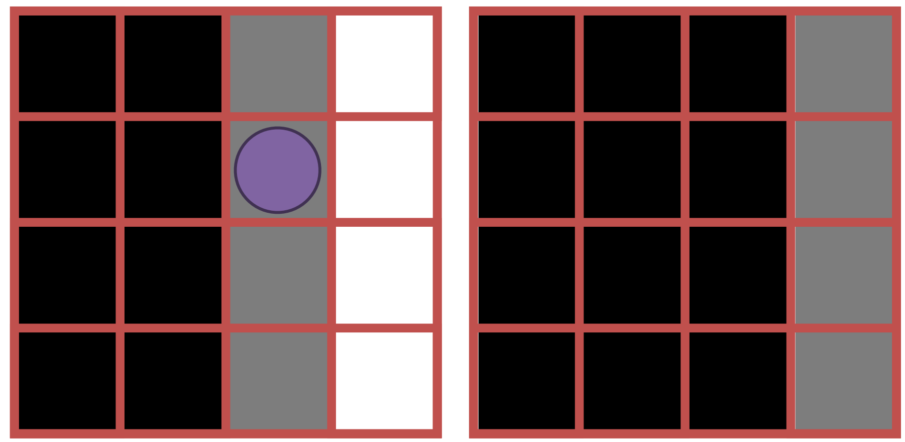


t   t+1

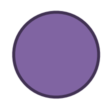
@ 

Ix = 1-0.5 = 0.5
Iy = 0
It = 1-0.5 = 0.5

What's u?



t   t+1

@ 

Ix = 1-0.5 = 0.5
Iy = 0
It = 0-0.5 = -0.5

What's u?

Adapted from David Fouhey

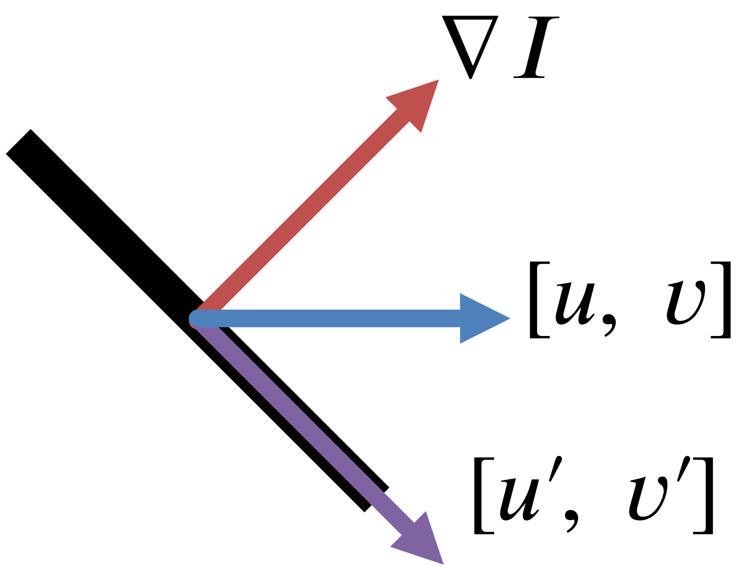
Optical Flow Equation

Have: $I_x u + I_y v + I_t = 0$ $I_t + \nabla I \cdot [u, v] = 0$

How many equations and unknowns per pixel?

1 (single equation), 2 (u and v)

One nasty problem:



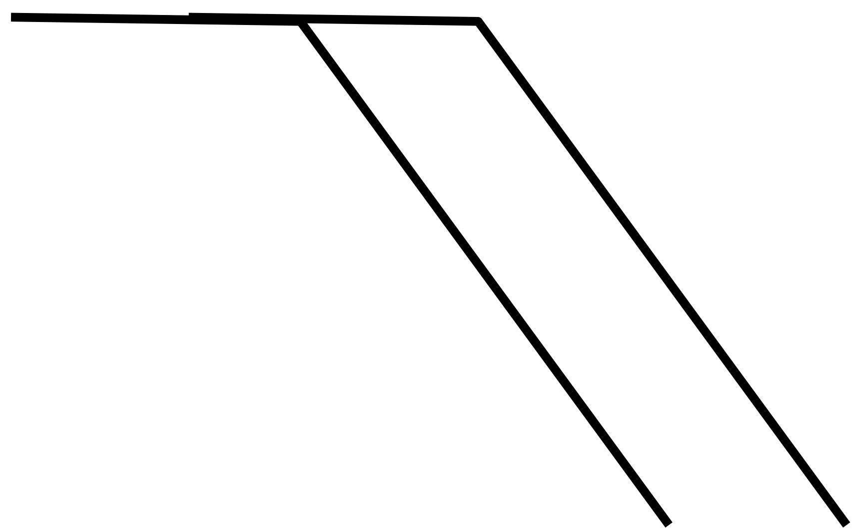
Suppose $\nabla I^T [u', v'] = 0$

$I_t + \nabla I^T [u + u', v + v'] = 0$

Can only identify the motion along gradient and **not** motion perpendicular to it

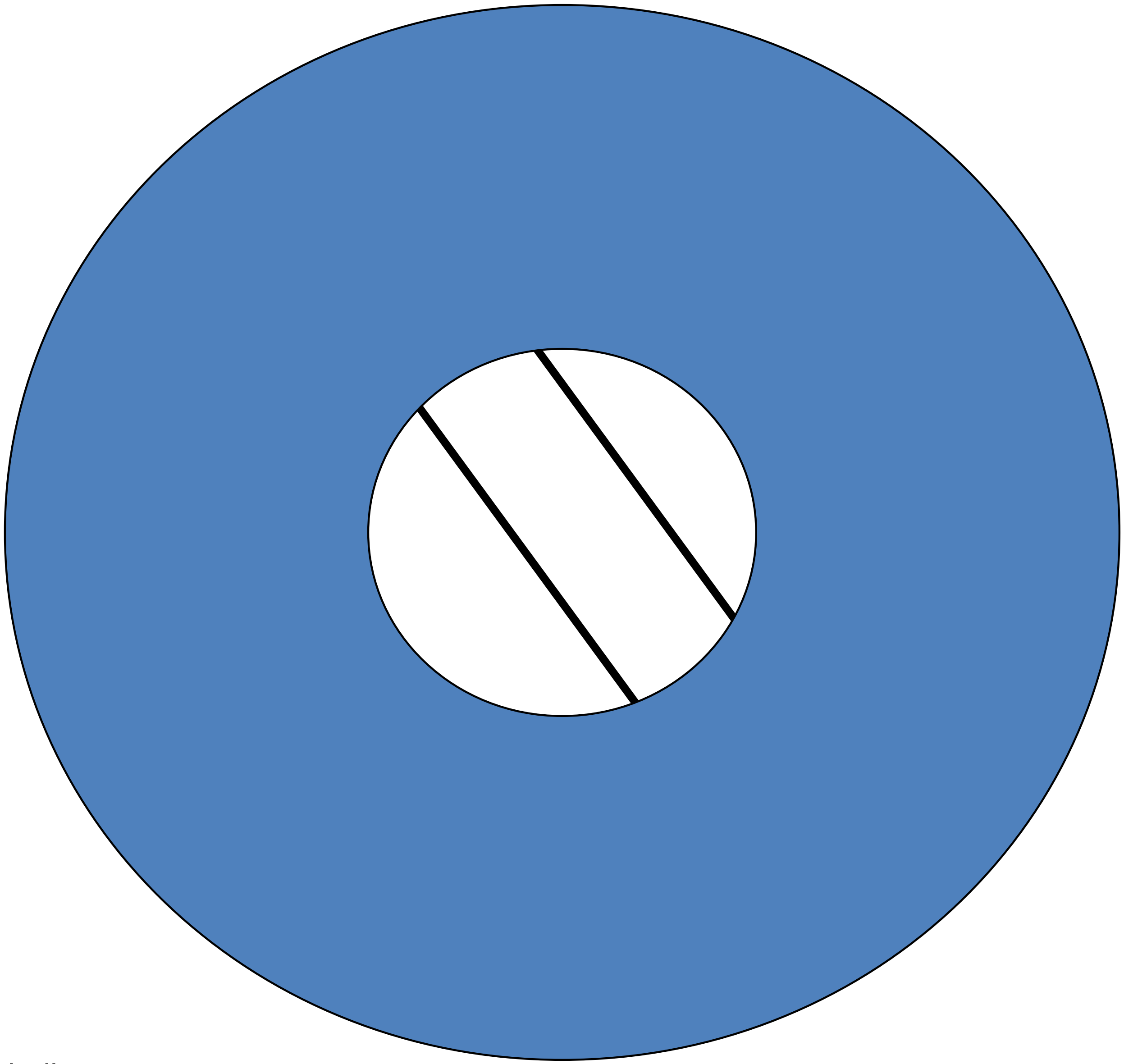
Adapted from S. Lazebnik slides

Aperture problem



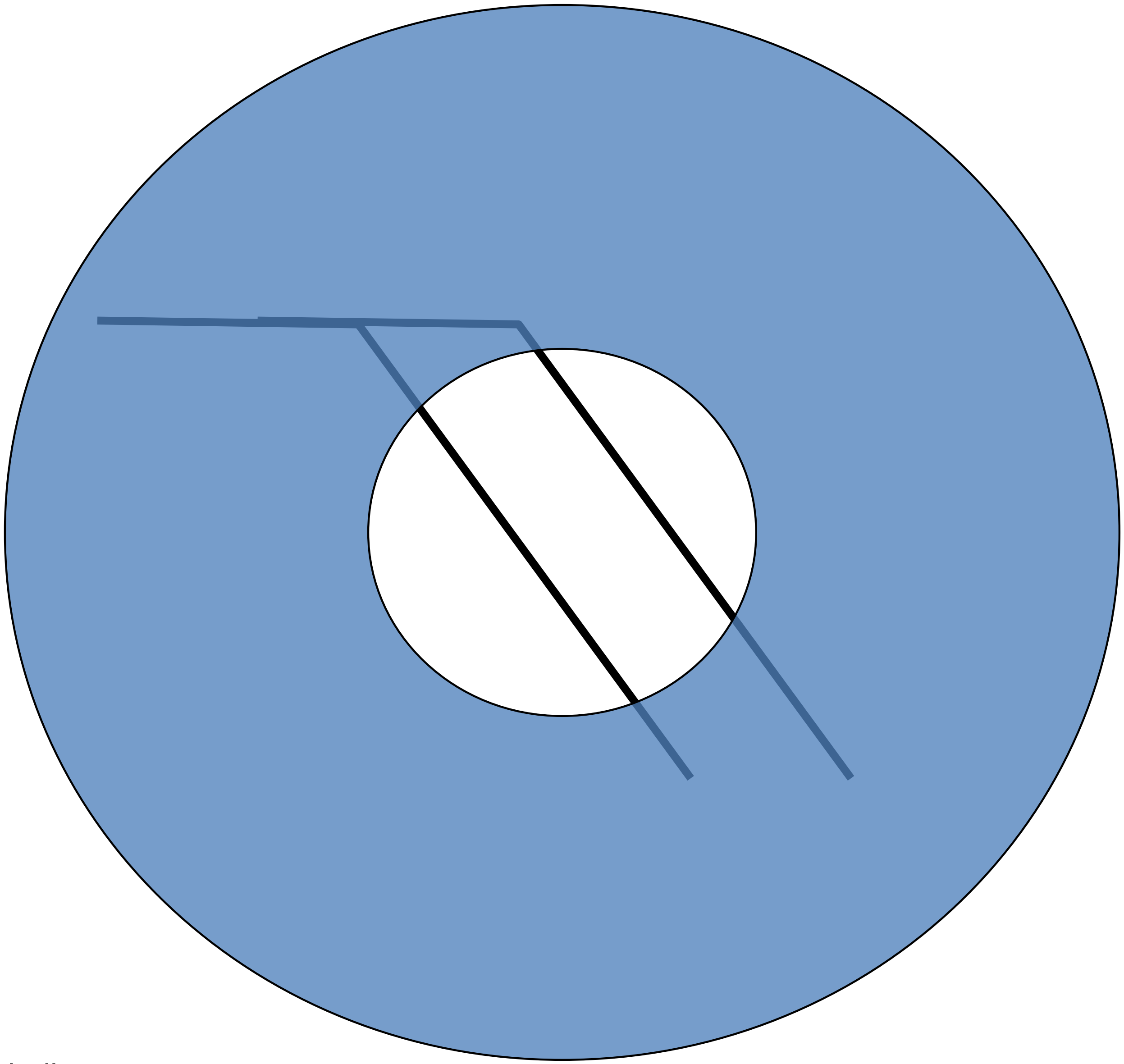
Slide credit: S. Lazebnik

Aperture problem



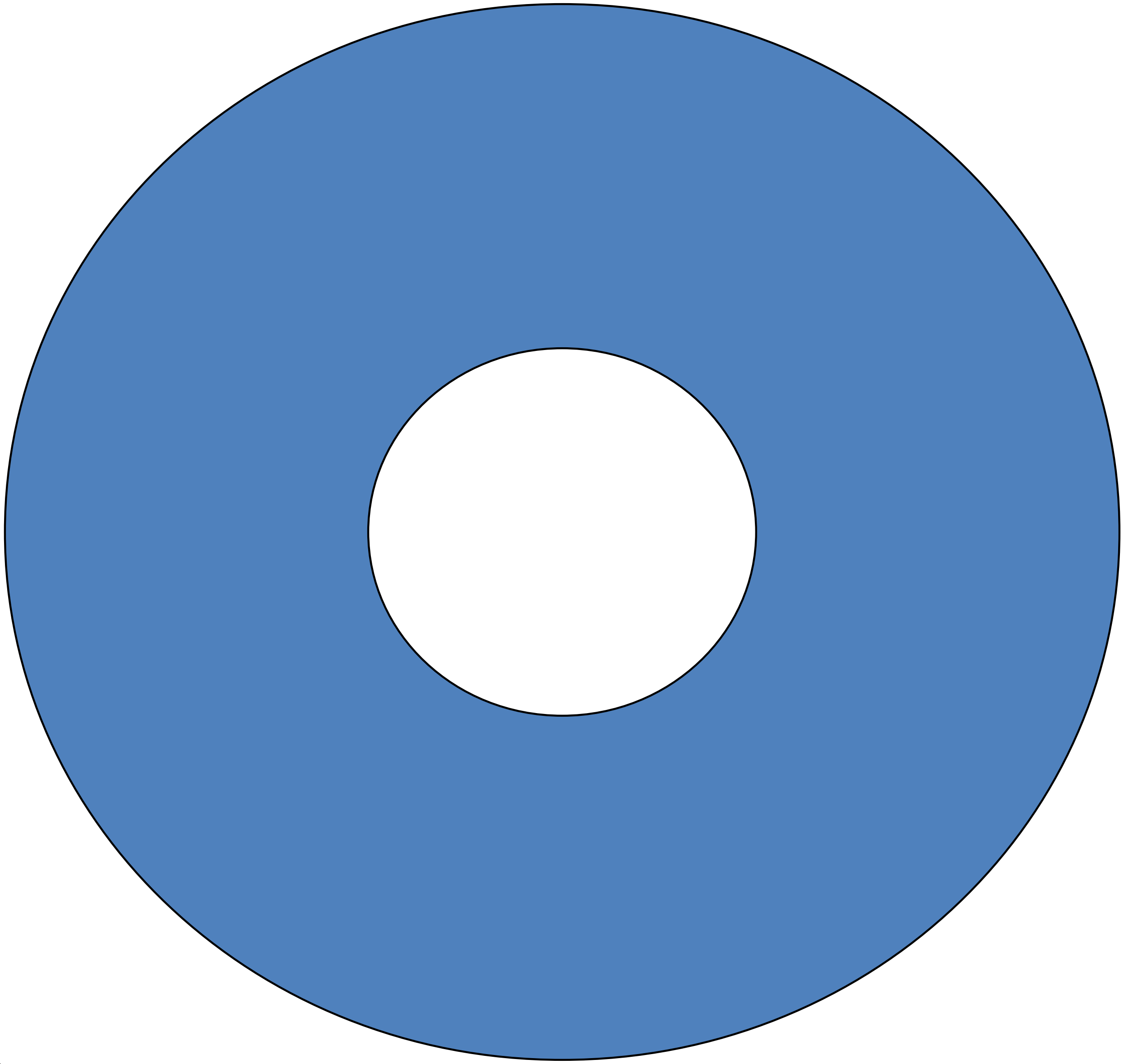
Slide credit: S. Lazebnik

Aperture problem



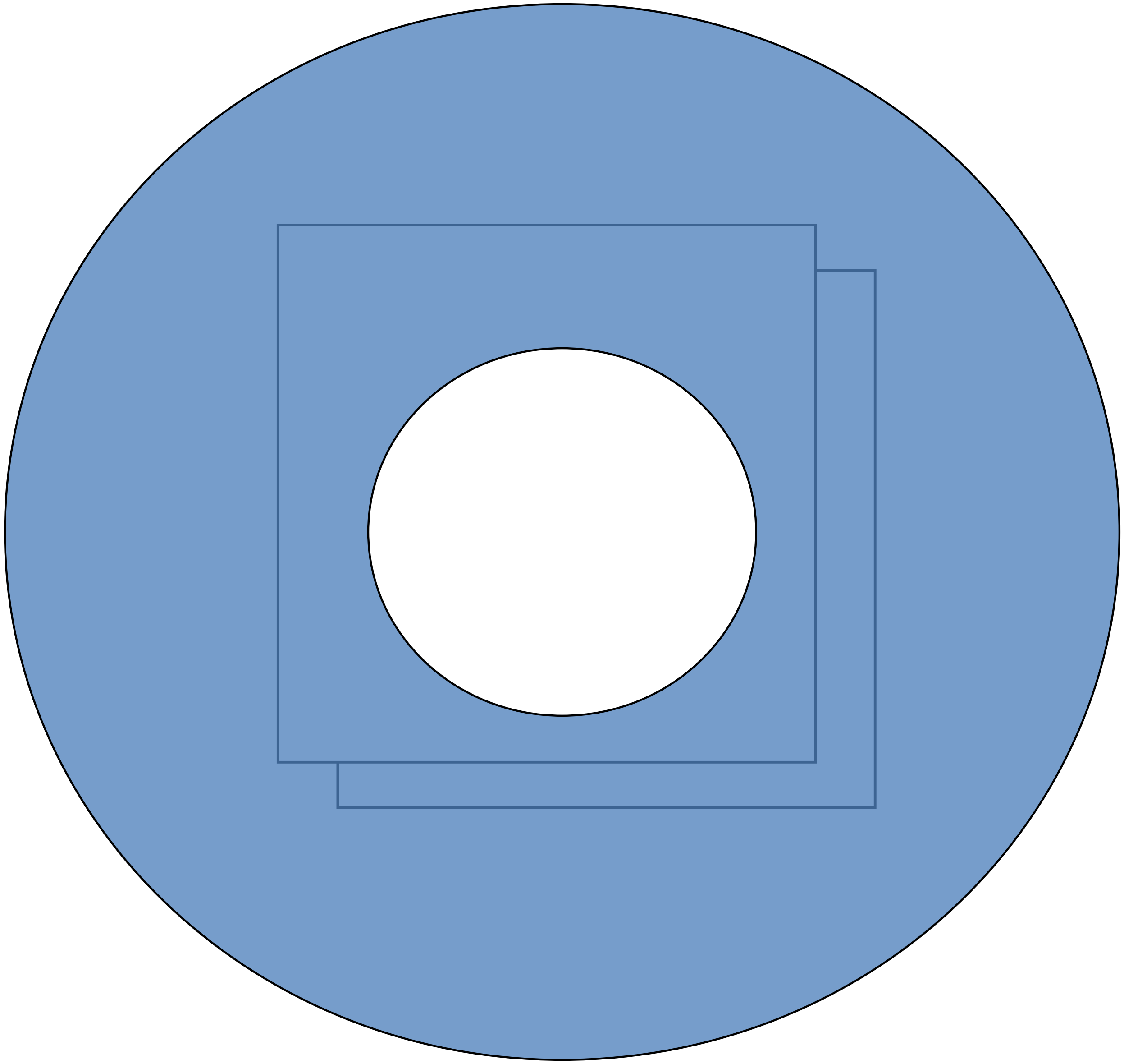
Slide credit: S. Lazebnik

Other Invisible Flow



Slide credit: D. Fouhey

Other Invisible Flow



Slide credit: D. Fouhey

Solving Ambiguity – Lucas Kanade

2 unknowns [u,v], 1 eqn per pixel
How do we get more equations?
Assume *spatial coherence*: pixel's neighbors have *move together* / have same [u,v]
5x5 window gives 25 new equations

$$I_t + I_x u + I_y v = 0$$

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}$$

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

Solving for [u,v]

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \qquad \underset{25x2}{\boldsymbol{A}} \; \underset{2x1}{\boldsymbol{d}} = \underset{25x1}{\boldsymbol{b}}$$

What's the solution?

$$\left(\boldsymbol{A}^T\boldsymbol{A}\right)\boldsymbol{d} = \boldsymbol{A}^T\boldsymbol{b} \quad \rightarrow \quad \boldsymbol{d} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}$$

Intuitively, need to solve (sum over pixels in window)

$$\underbrace{\begin{bmatrix} \sum \boldsymbol{I}_x\boldsymbol{I}_x & \sum \boldsymbol{I}_x\boldsymbol{I}_y \\ \sum \boldsymbol{I}_x\boldsymbol{I}_y & \sum \boldsymbol{I}_y\boldsymbol{I}_y \end{bmatrix}}_{\boldsymbol{A}^T\boldsymbol{A}} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum \boldsymbol{I}_x\boldsymbol{I}_t \\ \sum \boldsymbol{I}_y\boldsymbol{I}_t \end{bmatrix}}_{\boldsymbol{A}^T\boldsymbol{b}}$$

Adapted from S. Lazebnik slides

Solving for [u,v]

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

What does this remind you of?

Harris corner detection!

When can we find [u,v]?

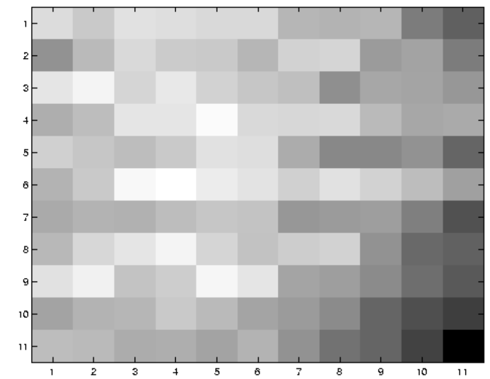
$\mathbf{A}^T\mathbf{A}$ invertible: precisely equal brightness

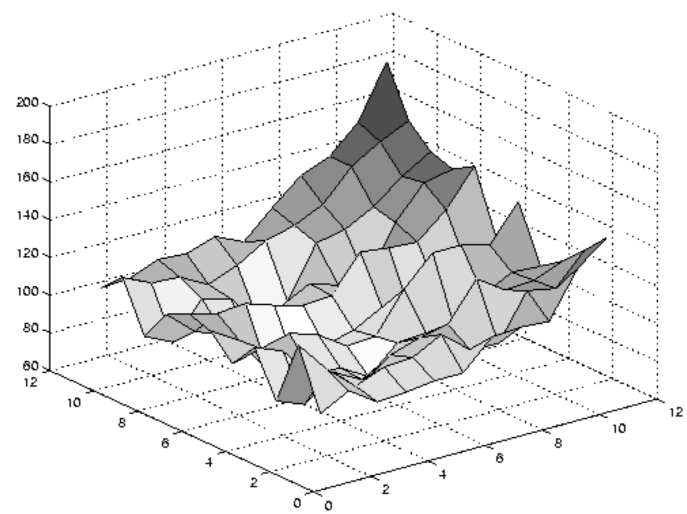
$\mathbf{A}^T\mathbf{A}$ not too small: noise + equal brightness

$\mathbf{A}^T\mathbf{A}$ well-conditioned: $|\lambda_1|/ |\lambda_2|$ not large (edge)

Adapted from S. Lazebnik slides

Low texture region





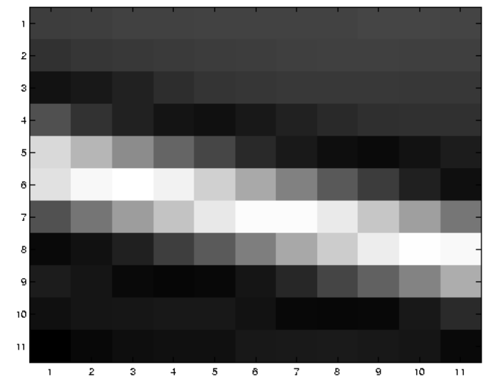
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \nabla I (\nabla I)^T$$

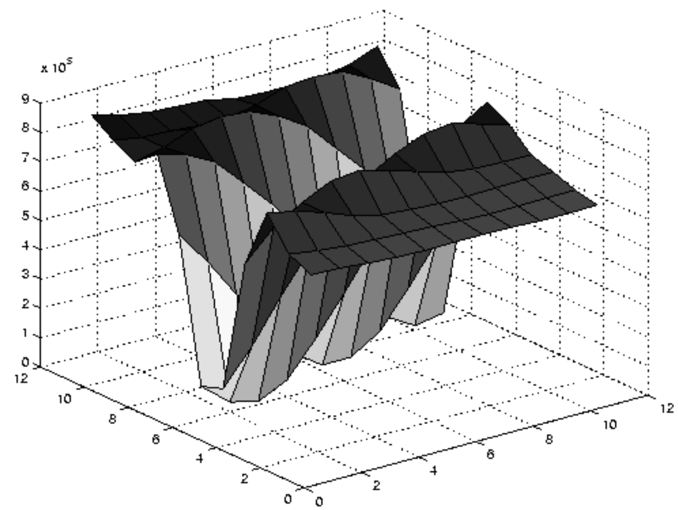
– gradients have small magnitude

– small λ_1, small λ_2

Slide credit: S. Lazebnik

Edge





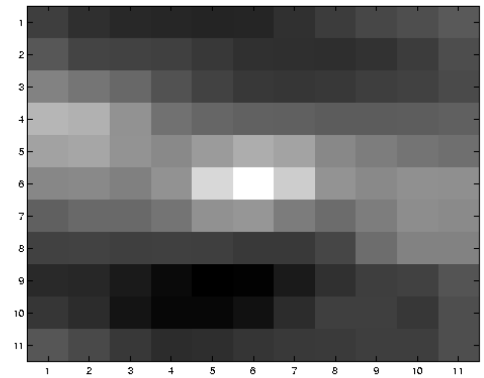
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \nabla I (\nabla I)^T$$

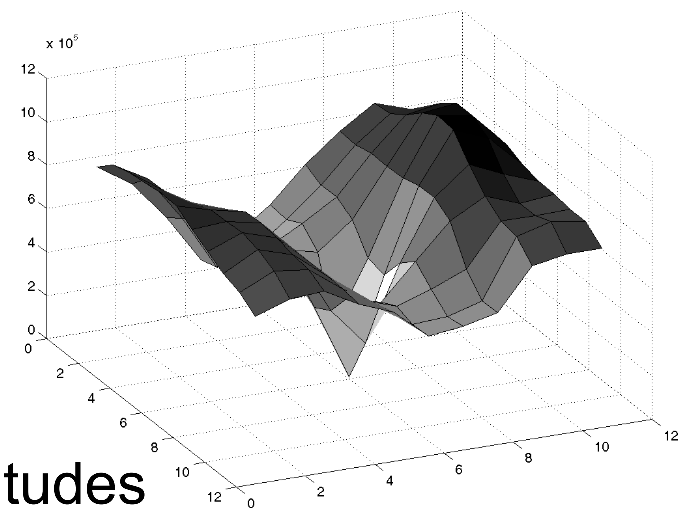
– large gradients, all the same

– large λ_1, small λ_2

Slide credit: S. Lazebnik

High texture region





$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

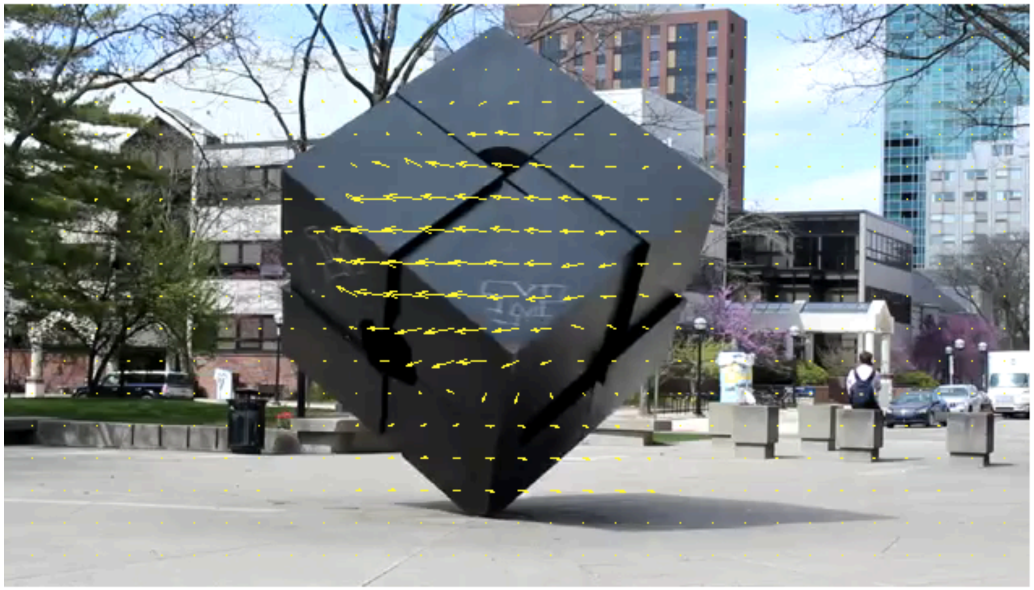
– large λ_1, large λ_2

Slide credit: S. Lazebnik

Lucas-Kanade flow example

Input frames

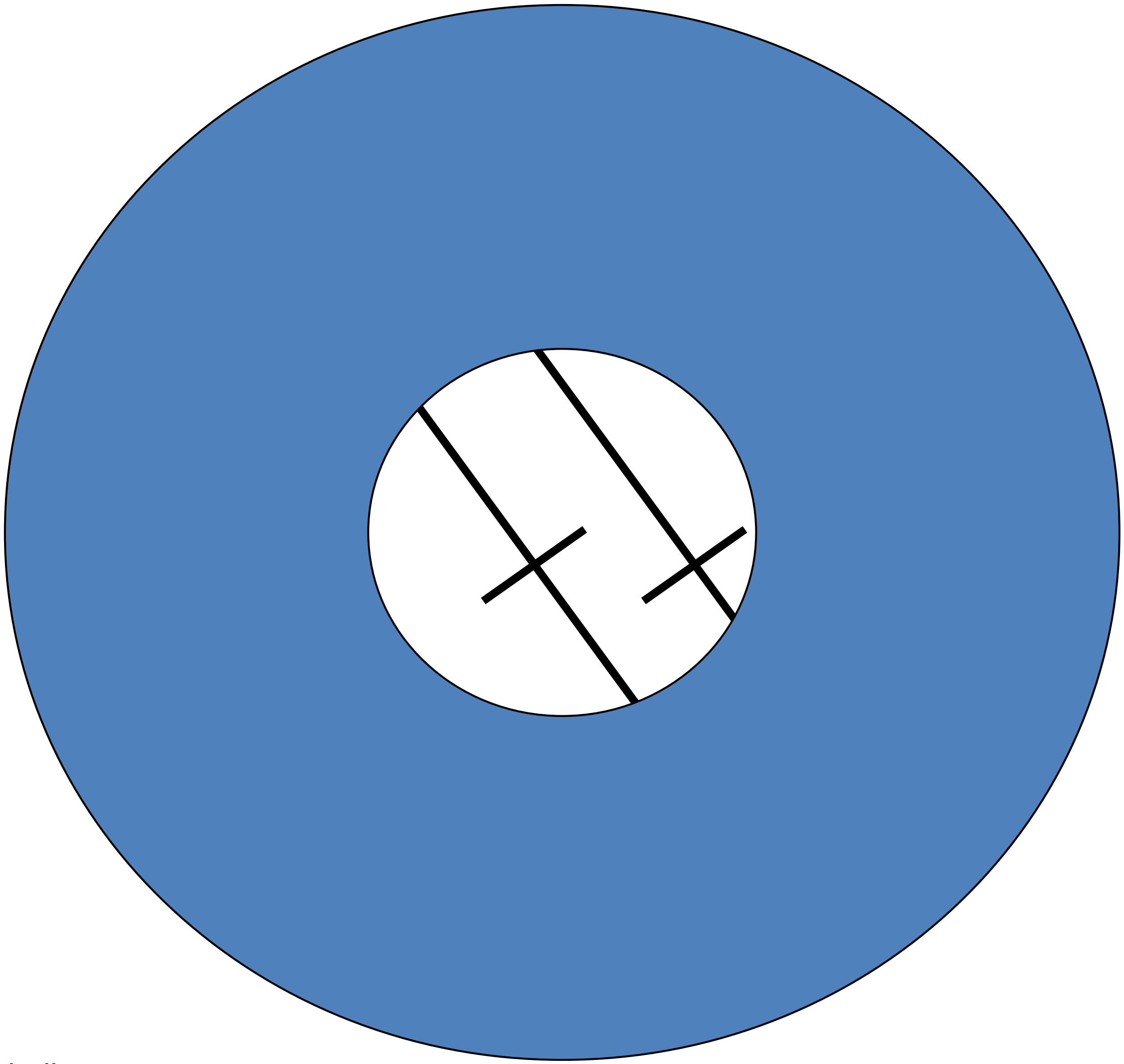
Output



Slide credit: S. Lazebnik

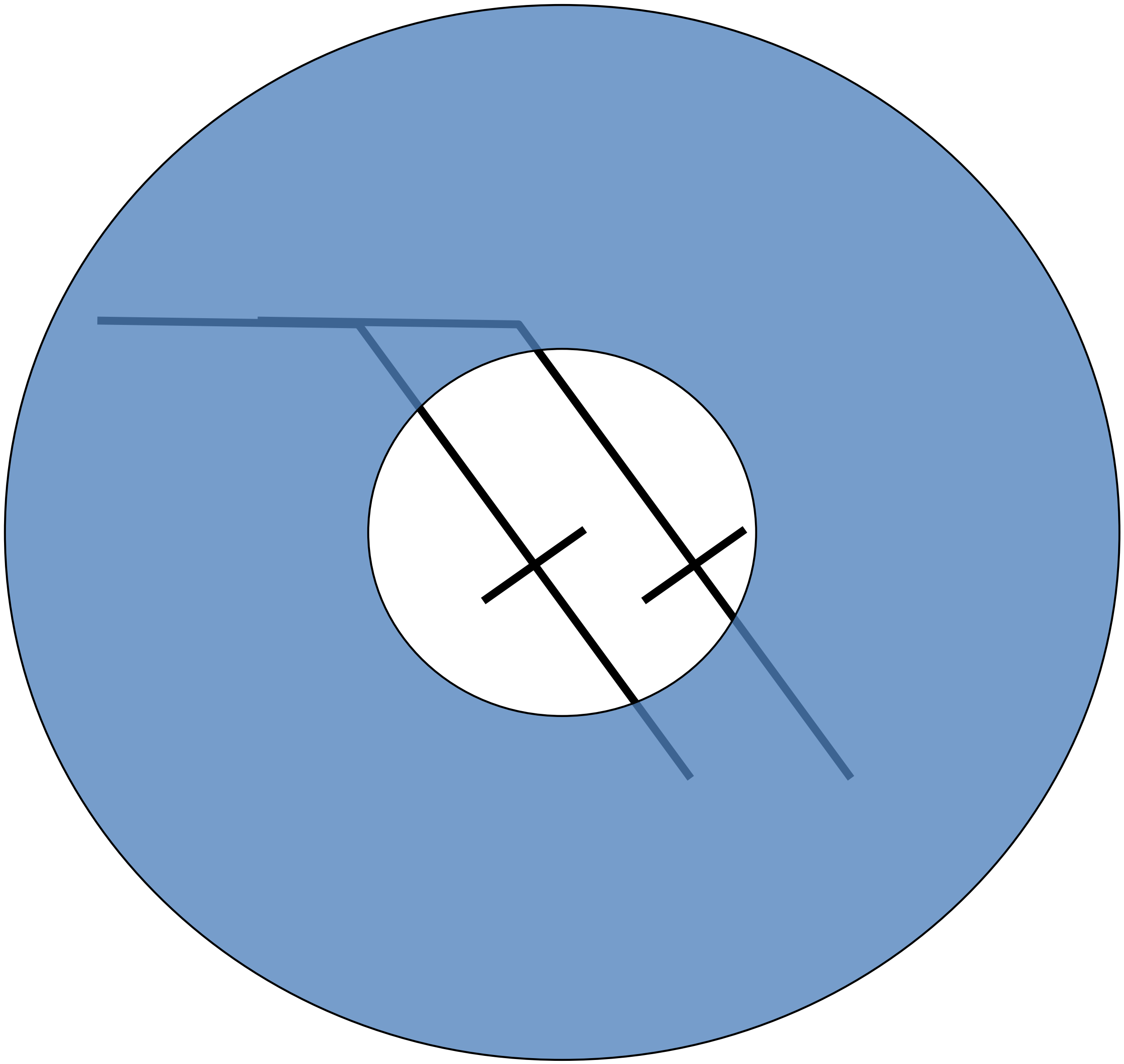
Source: [MATLAB Central File Exchange]

Aperture problem Take 2



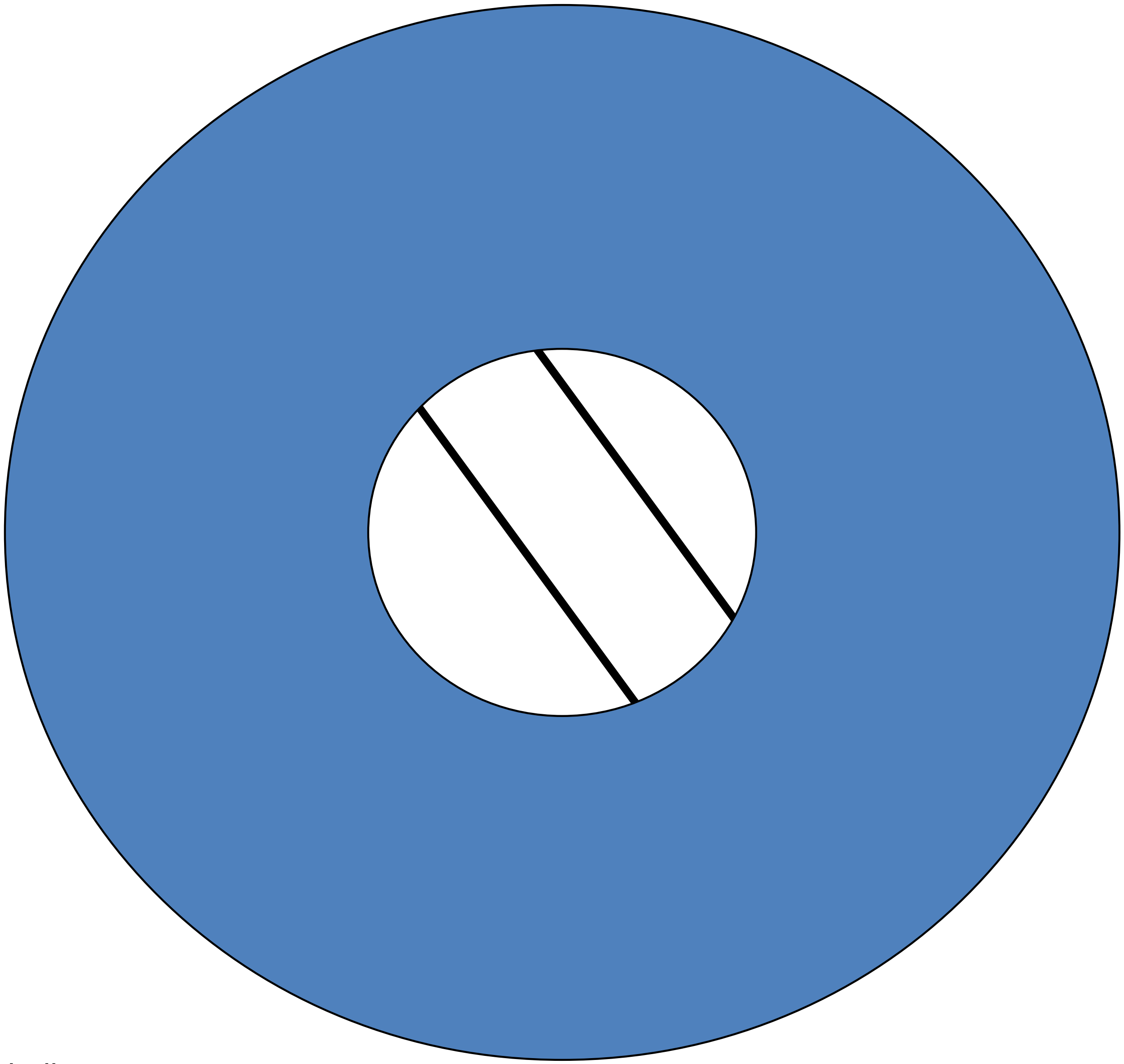
Slide credit: S. Lazebnik

Aperture problem Take 2



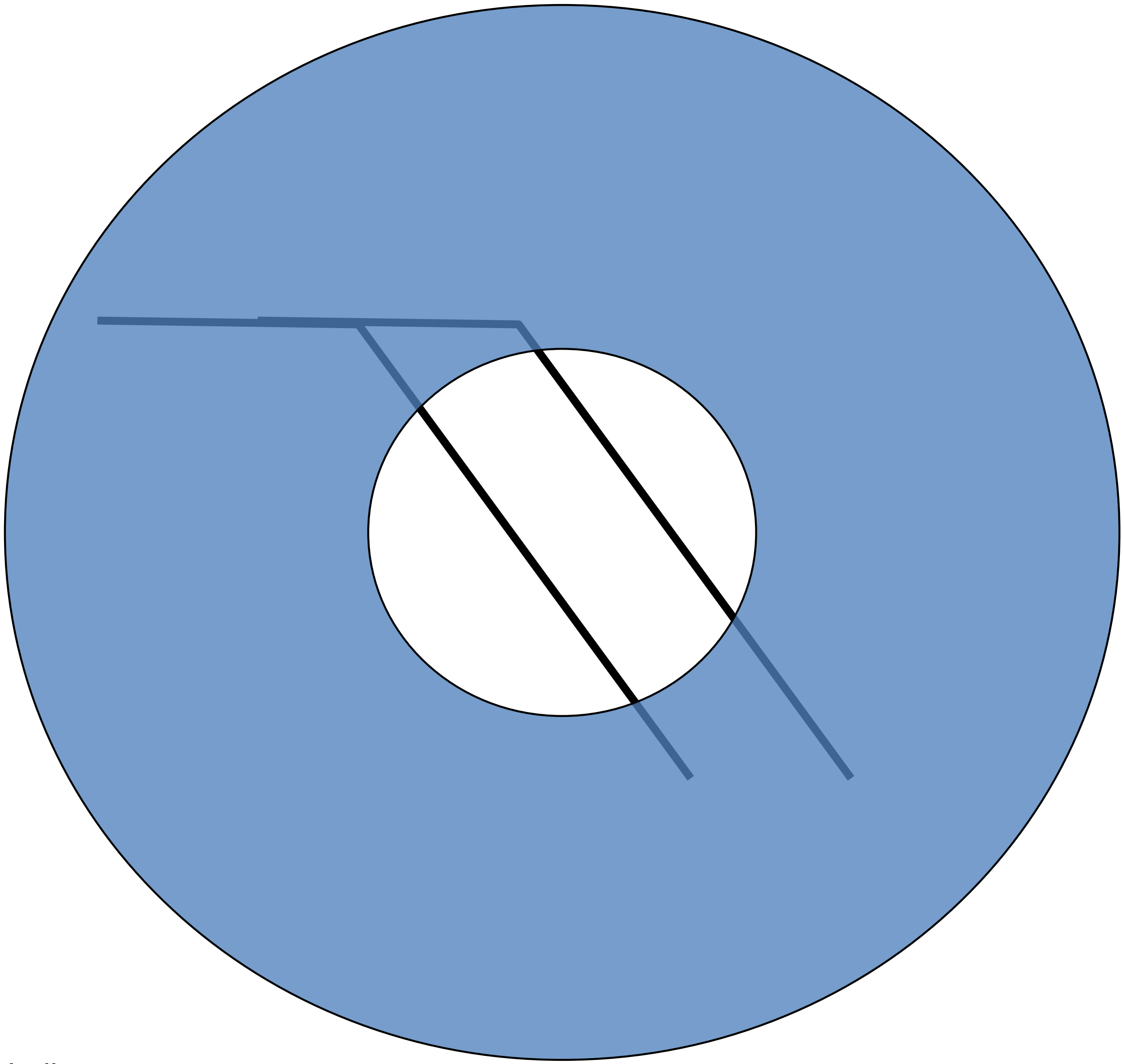
Slide credit: S. Lazebnik

For Comparison



Slide credit: S. Lazebnik

For Comparison



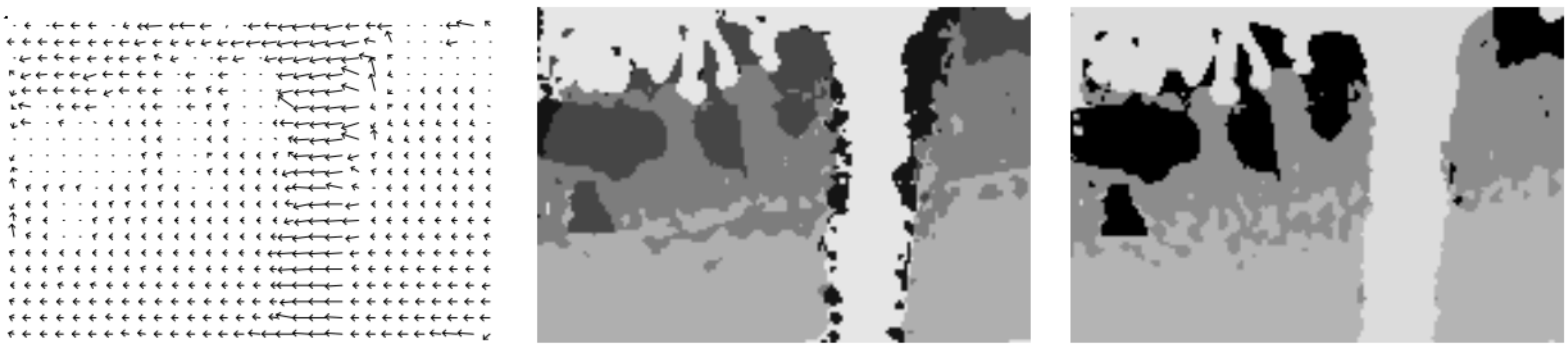
Slide credit: S. Lazebnik

So How Does This Fail?

- Point doesn't move like neighbors:
  - **Why would this happen?**
  - Figure out which points move together, then come back and fix.

So How Does This Fail?

- Point doesn't move like neighbors:
  - **Why would this happen?**
  - Figure out which points move together, then come back and fix



(a) (b) (c)

Figure 11: (a) The optic flow from multi-scale gradient method. (b) Segmentation obtained by clustering optic flow into affine motion regions. (c) Segmentation from consistency checking by image warping. Representing moving images with layers.

J. Wang and E. Adelson, Representing Moving Images with Layers, IEEE Transactions on Image Processing, 1994

So How Does This Fail?

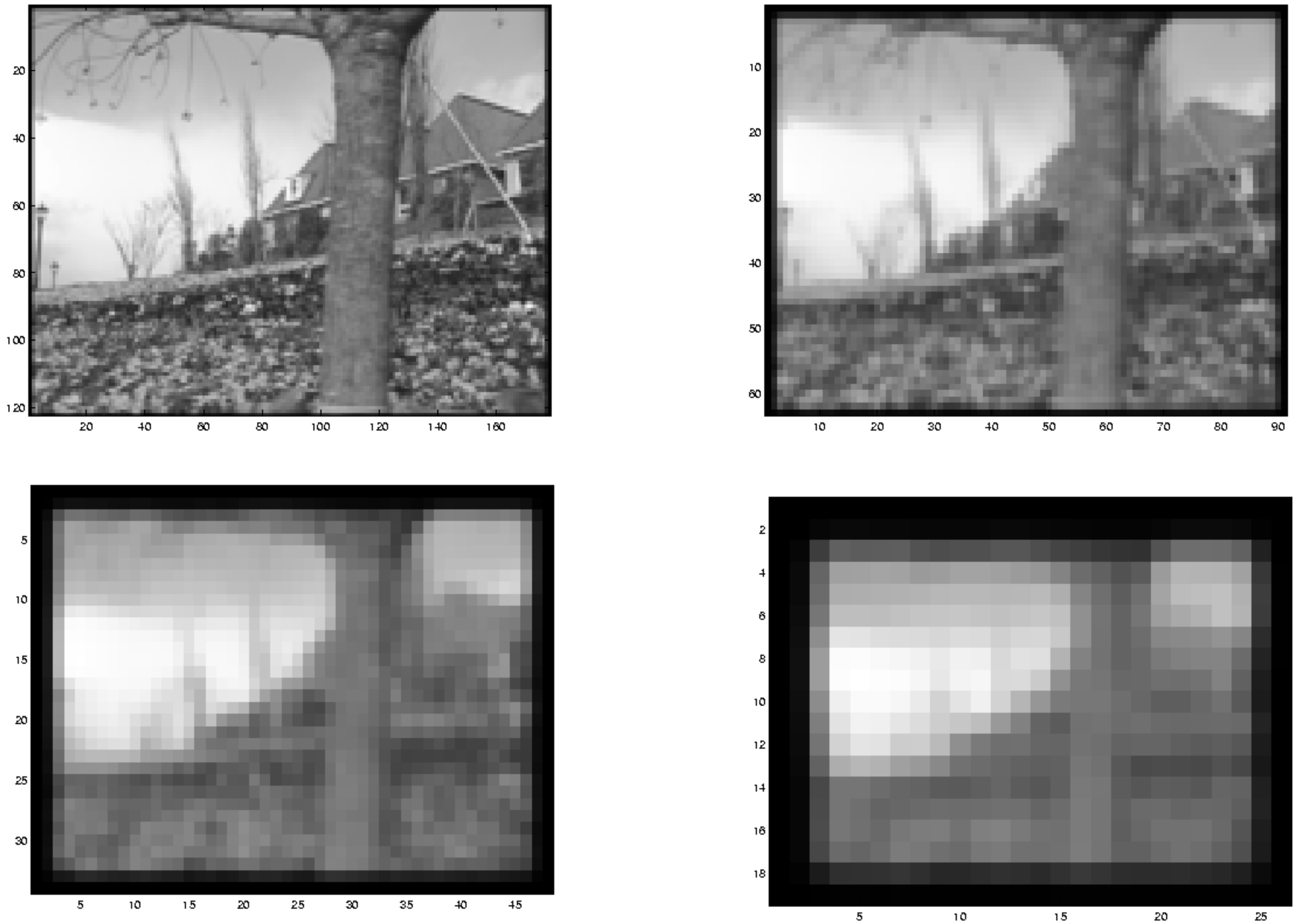
- Point doesn't move like neighbors:
  - **Why would this happen?**
  - Figure out which points move together, then come back and fix
- Brightness constancy isn't true
  - **Why would this happen?**
  - Solution: other form of matching (e.g. SIFT)
- Taylor series is bad approximation
  - **Why would this happen?**
  - Solution: Make your pixels big

Revisiting small motions

- Is this motion small enough?
  - Probably not—it's much larger than one pixel
  - How might we solve this problem?

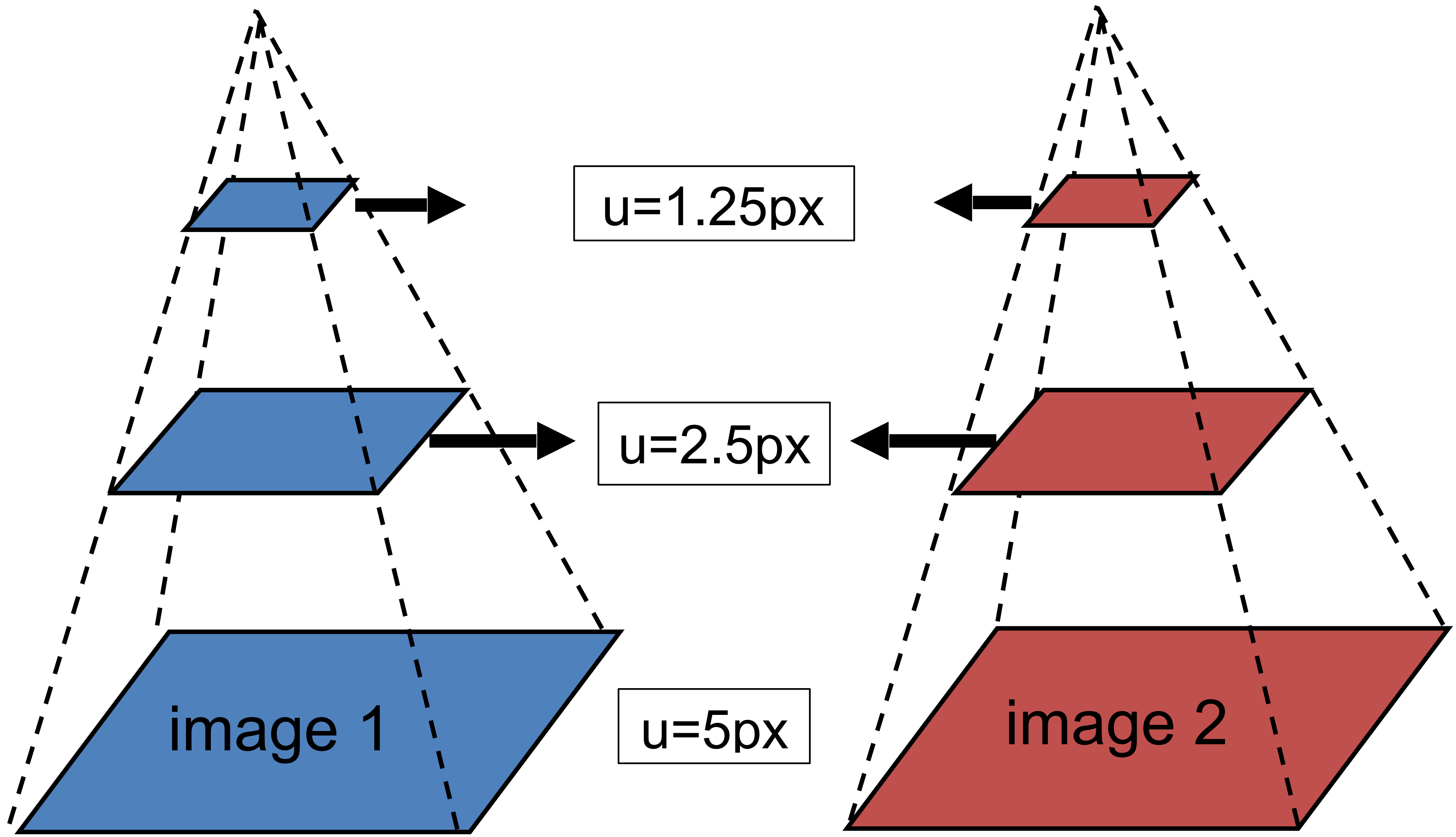
Slide credit: S. Lazebnik

Reduce the resolution!



Slide credit: S. Lazebnik

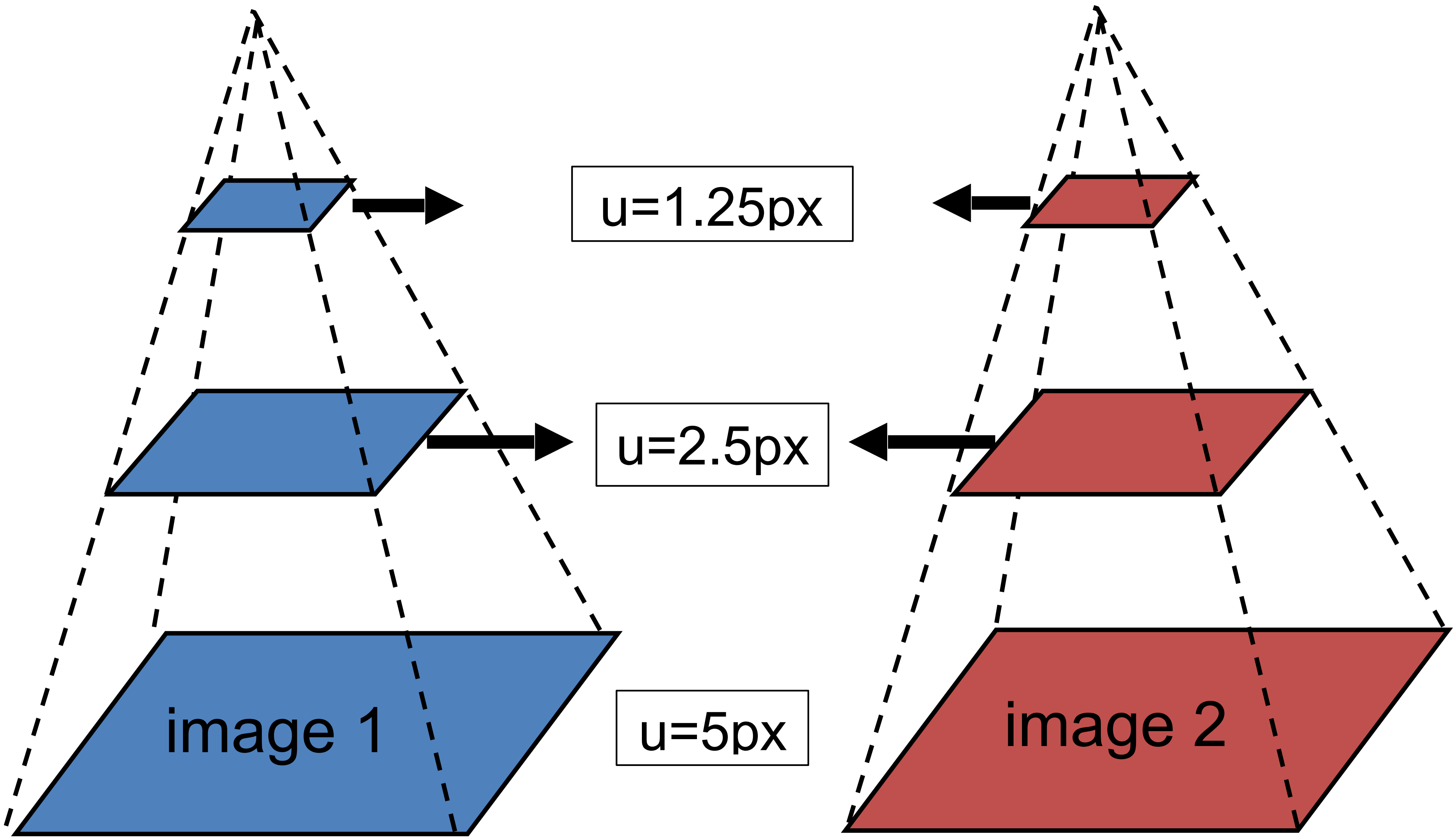
Coarse-to-fine optical flow estimation



Typically called Gaussian Pyramid

Slide credit: S. Lazebnik

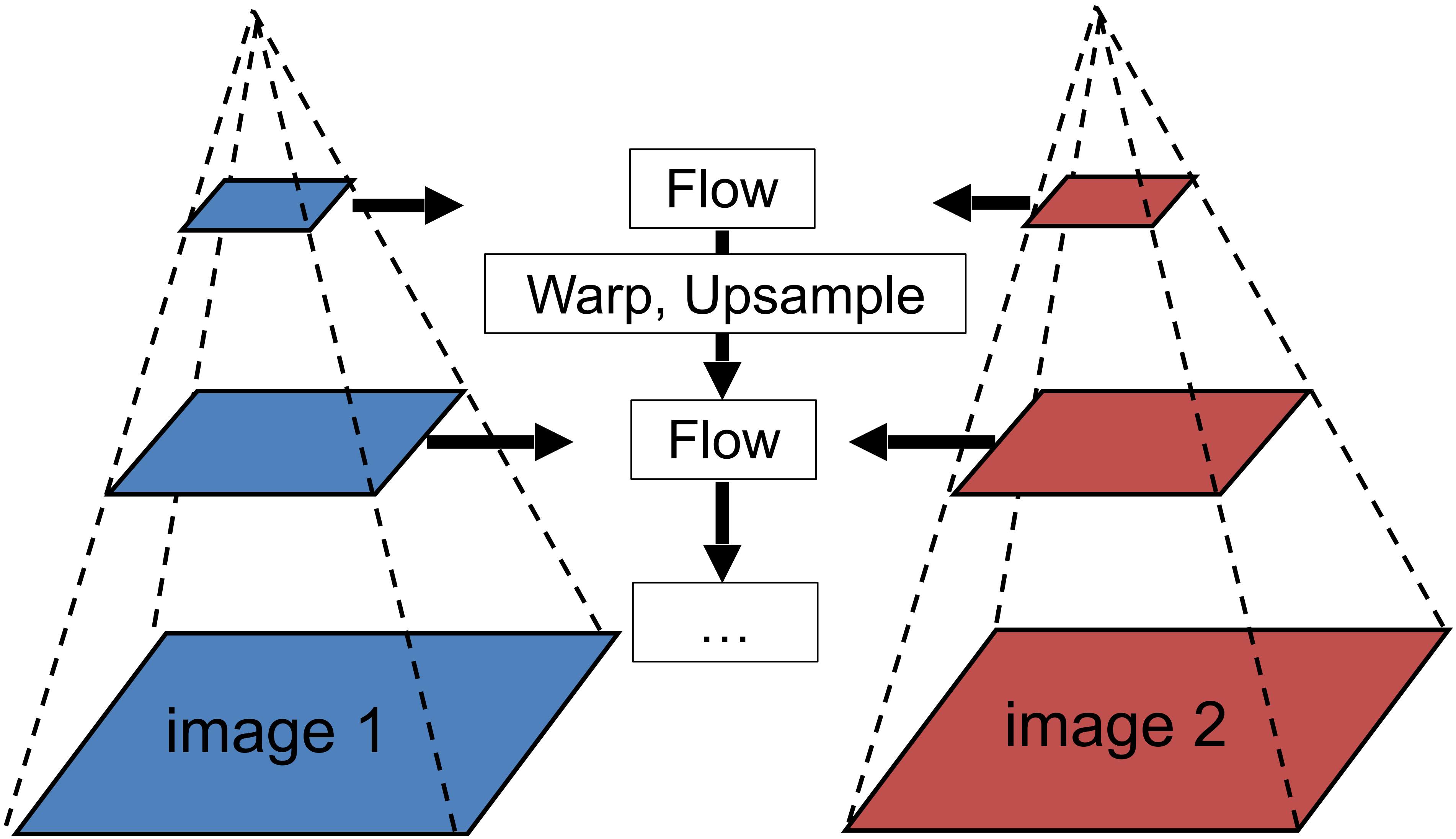
Coarse-to-fine optical flow estimation



Do we start at bottom or top to align?

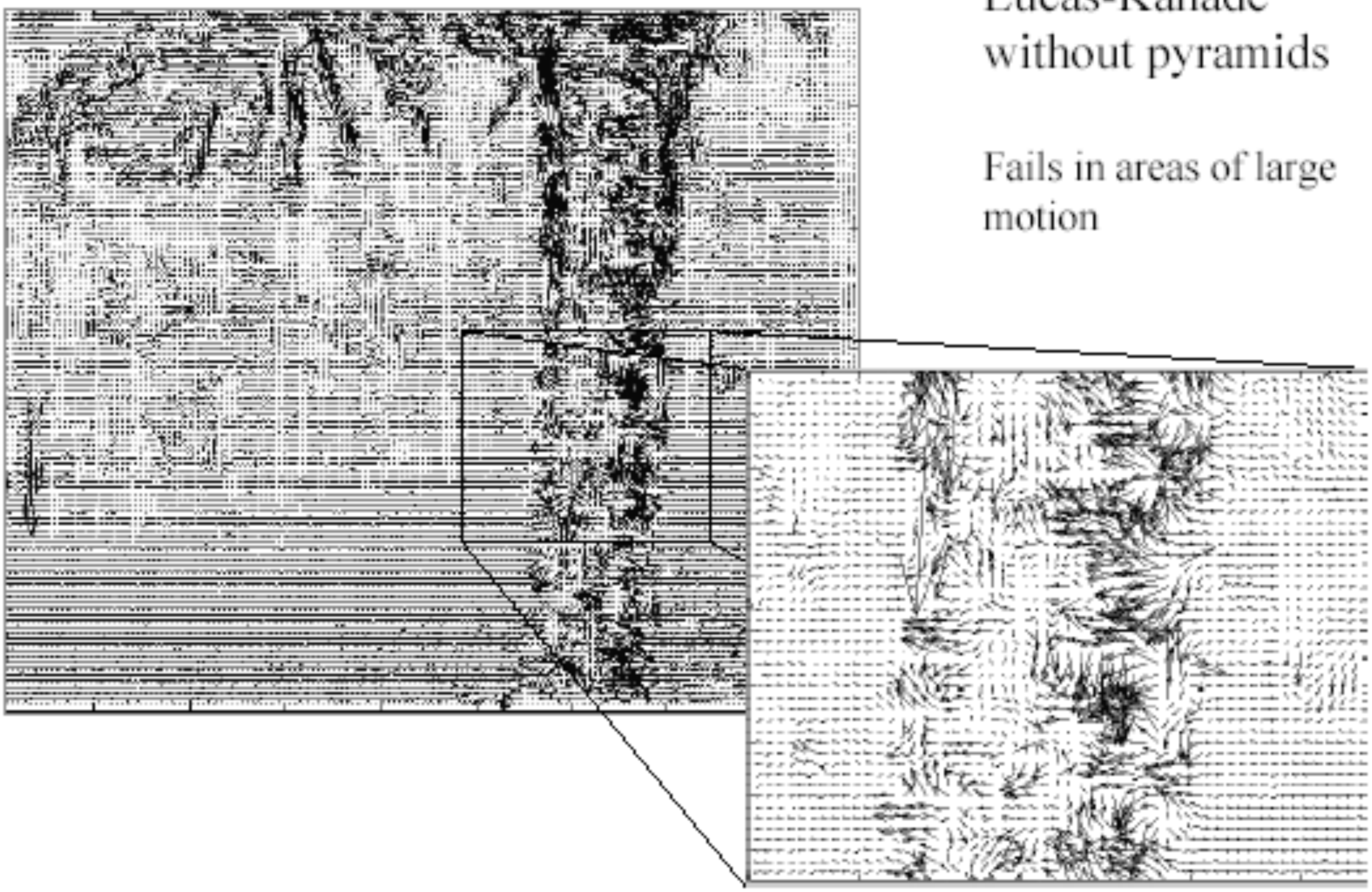
Slide credit: S. Lazebnik

Coarse-to-fine optical flow estimation



Slide credit: S. Lazebnik

Optical Flow Results

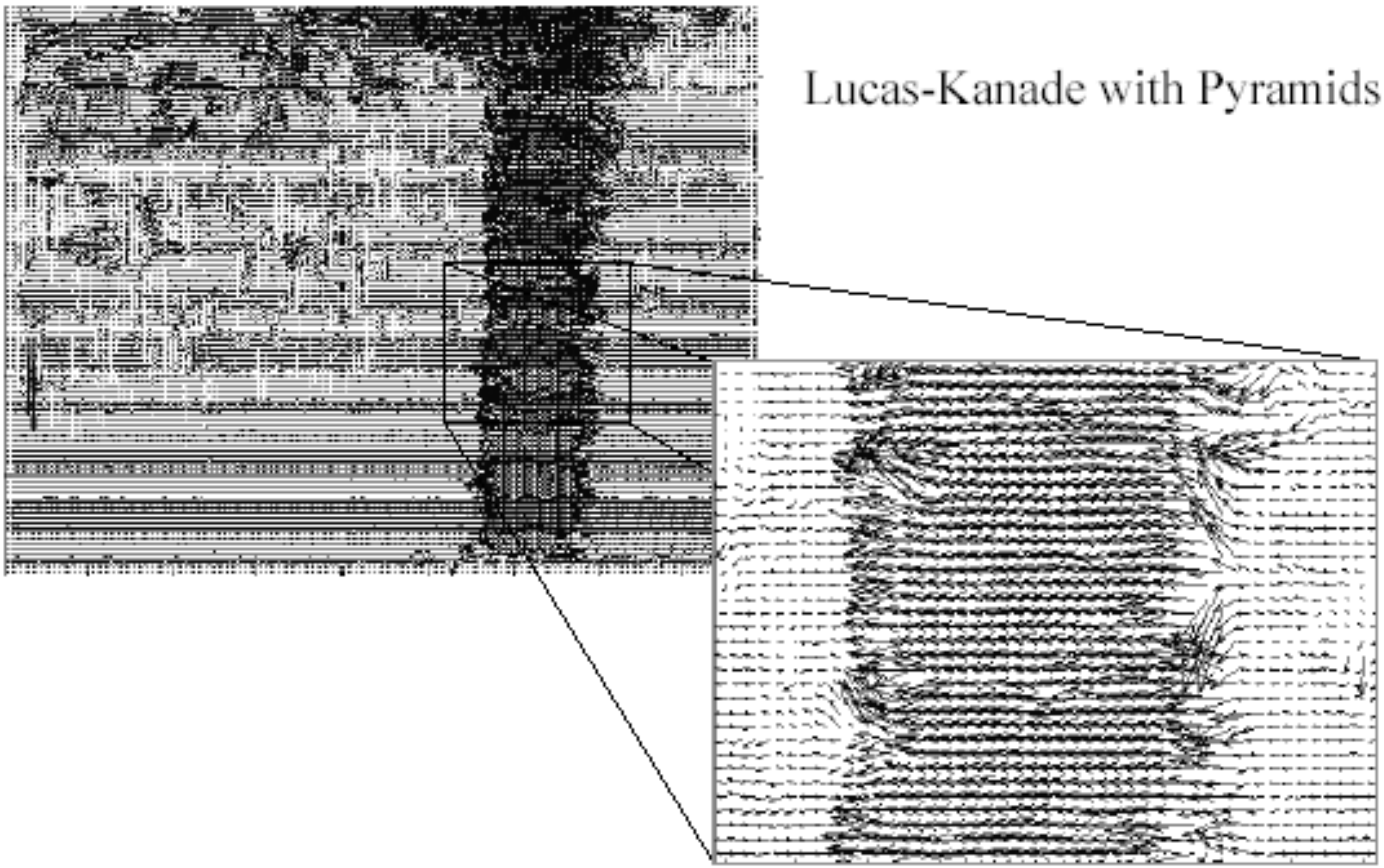


Lucas-Kanade
without pyramids

Fails in areas of large
motion

Slide credit: K. Hassan-Shafique

Optical Flow Results



Lucas-Kanade with Pyramids

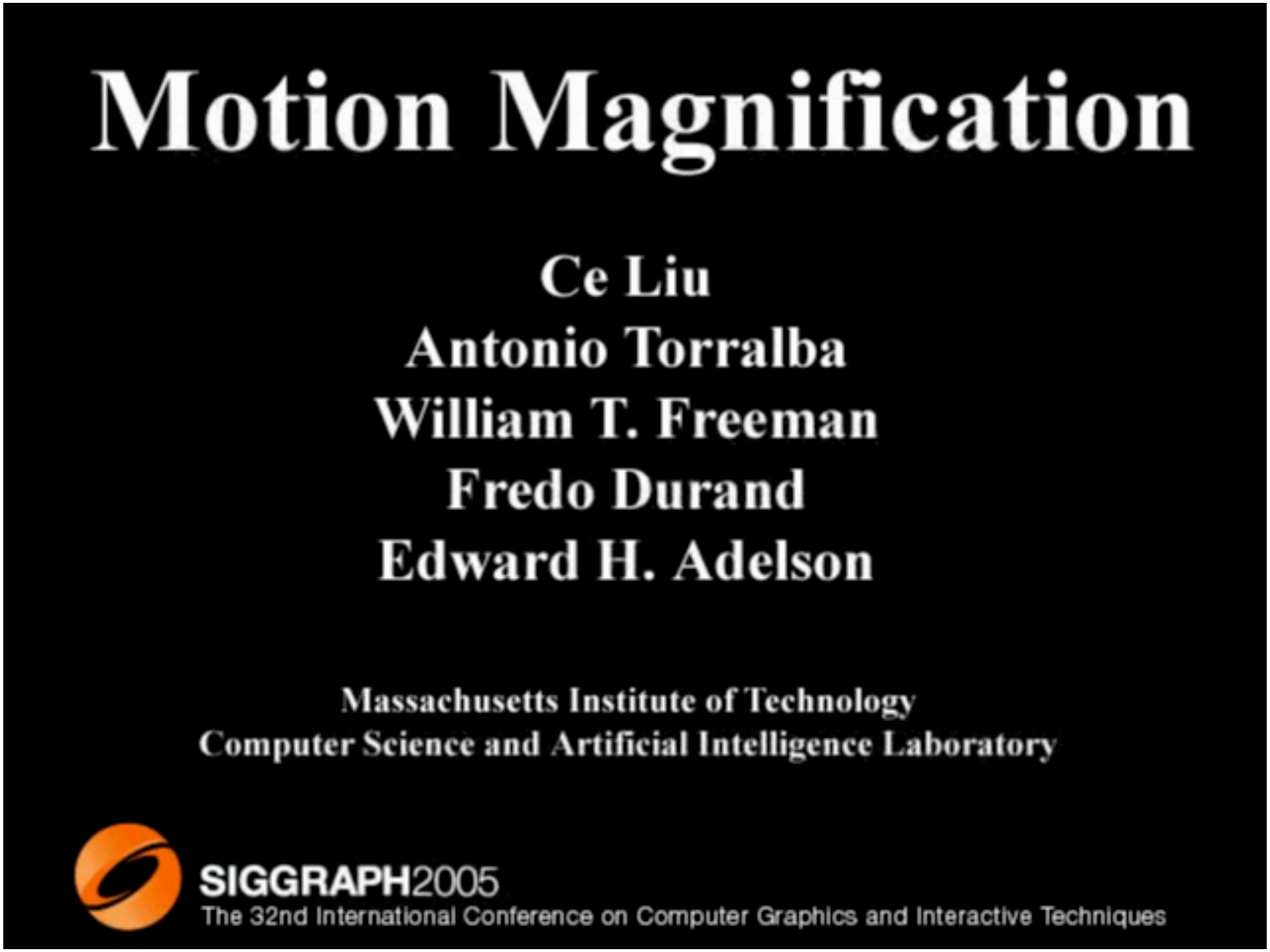
Slide credit: K. Hassan-Shafique

Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Ideally, optical flow would be the same as the motion field
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

Slide credit: S. Lazebnik

Motion Magnification



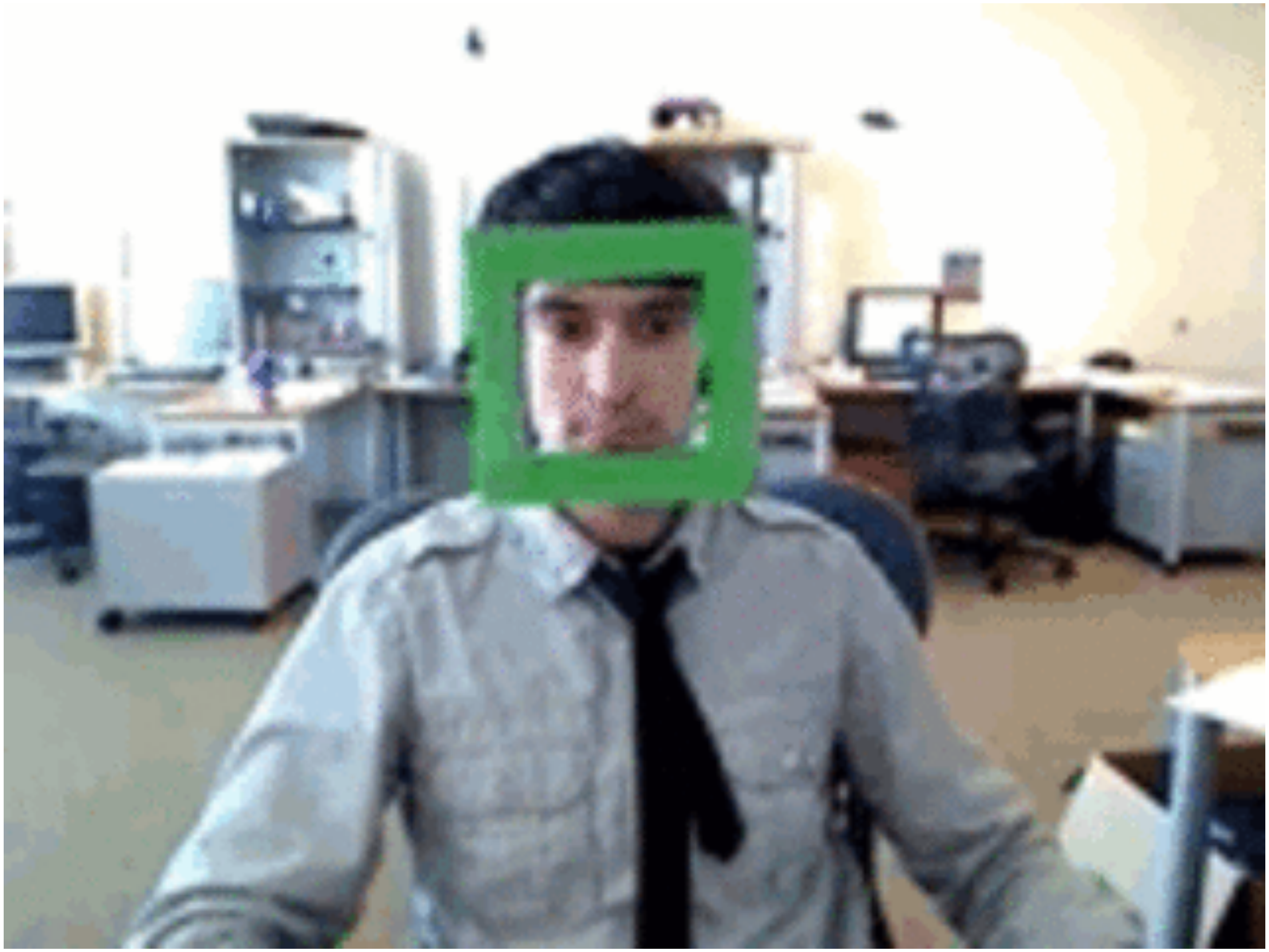
Basics of tracking objects

Tracking Examples

Video credit: B. Babenko

Tracking Examples

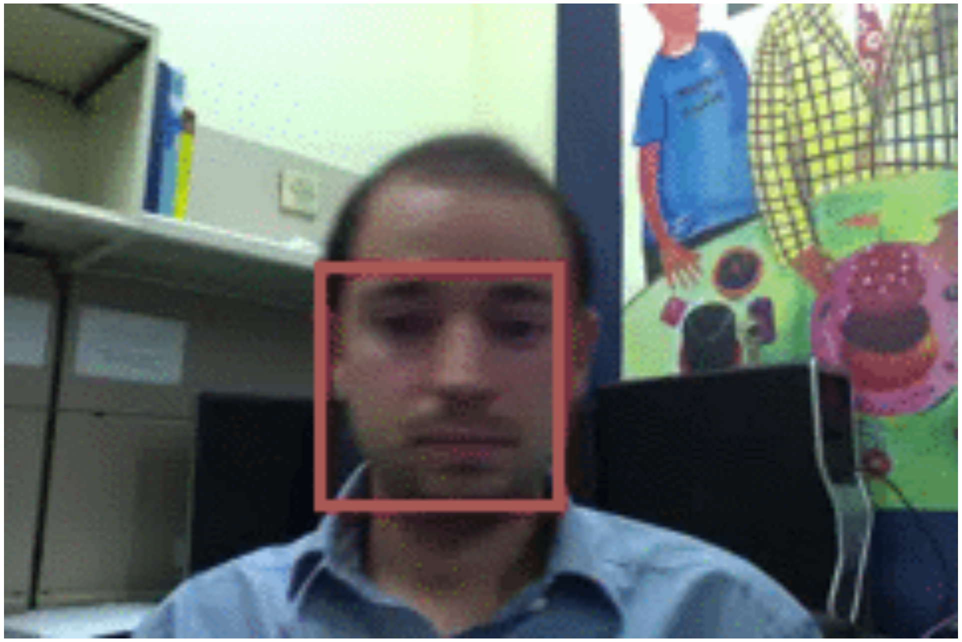
Best Tracking



Slide credit: B. Babenko

Difficulties

- Erratic movements, rapid motion
- Occlusion
- Surrounding similar objects



Slide credit: D. Hoiem

Tracking by Detection

Tracking by detection:

- Works if object is detectable
- Need some way to link up detections

Slide credit: D. Hoiem

Tracking With Dynamics

Based on motion, predict object location

- Restrict search for object
- Measurement noise is reduced by smoothness
- Robustness to missing or weak observations

Slide credit: D. Hoiem

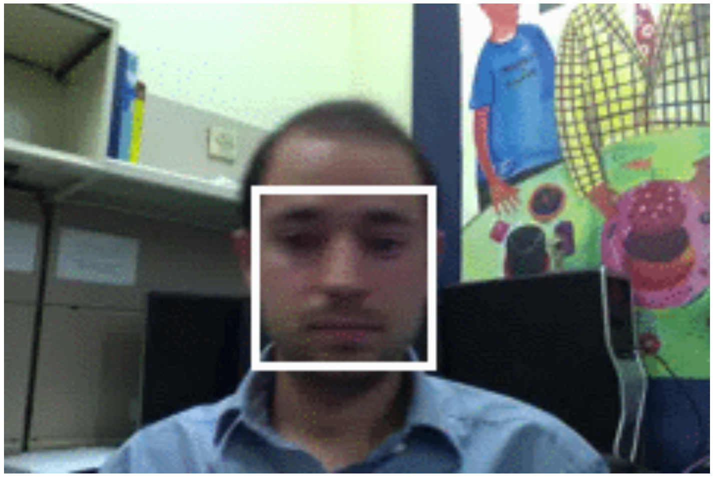
Strategies For Tracking

- Tracking with motion prediction:
  - Predict object's state in next frame.
  - Fuse with observation.

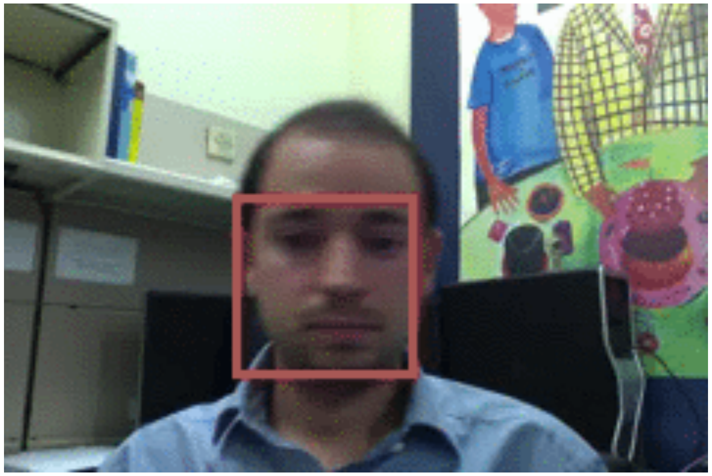
Slide credit: D. Hoiem

General Tracking Model

State X: actual state of object that we want to estimate. Could be: Pose, viewpoint, velocity, acceleration.



Observation Y: our "measurement" of state X. Can be noisy. At each time step t, state changes to X_t, get Y_t.



Slide credit: D. Hoiem

Probabilistic tracking

Have models for:

(1) P(next state) given current state / *Transition*

$$P(X_t \mid X_{t-1})$$

(2) P(observation) given state / *Observation*

$$P(Y_t \mid X_t)$$

Want to recover, for each timestep t

$$P(X_t \mid y_0, \ldots, y_t)$$

Slide credit: D. Hoiem

Probabilistic tracking

- Base case:
  - Start with initial ***prediction***/prior: $P(X_0)$
  - For the first frame, ***correct*** this given the first measurement: $Y_0 = y_0$

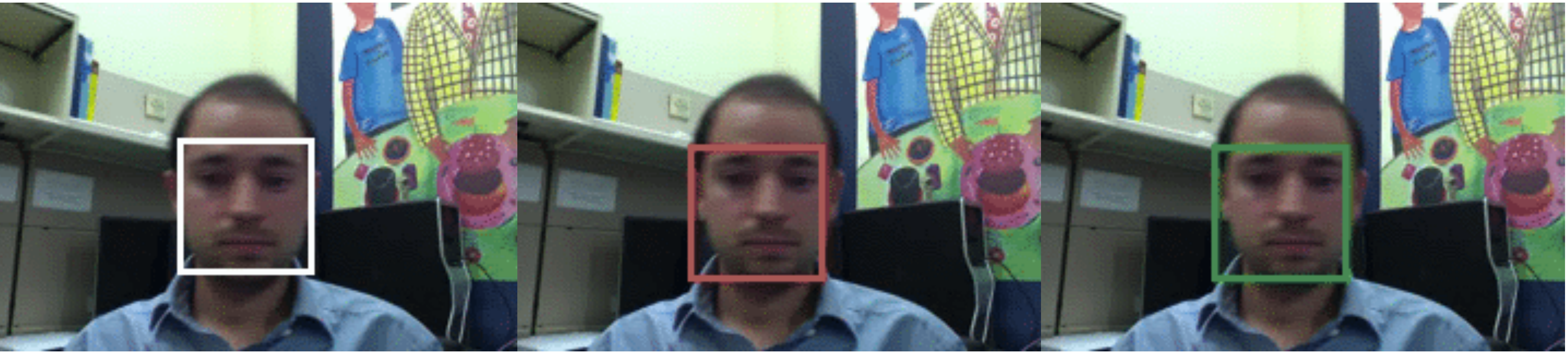
Slide credit: D. Hoiem

Probabilistic tracking

- Base case:
  - Start with initial ***prediction***/prior: $P(X_0)$
  - For the first frame, ***correct*** this given the first measurement: $Y_0=y_0$
- Each subsequent step:
  - ***Predict*** X_t given past evidence
  - Observe y_t: ***correct*** X_t given current evidence

Slide credit: D. Hoiem

Comparison



Ground Truth | Observation | Correction

Slide credit: D. Hoiem

Two common solutions

- Kalman filter:
  - Pretend everything is Gaussian
  - Keep track of mean, variance of estimated location
  - Very efficient
- Particle filter:
  - Represent the state distribution non-parametrically using K samples
  - Prediction: sample the next K possible locations $X_{k,t+1}$
  - Correction: compute likelihood of each $X_{k,t+1}$ based on observations

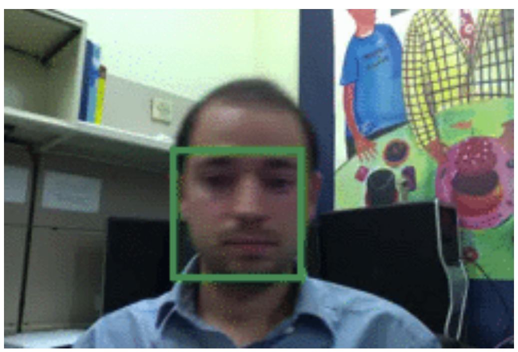
Tracking Issues

- Initialization
- Getting observation and dynamics models
  - Observation model: match template or use trained detector
  - Dynamics Model: specify with domain knowledge

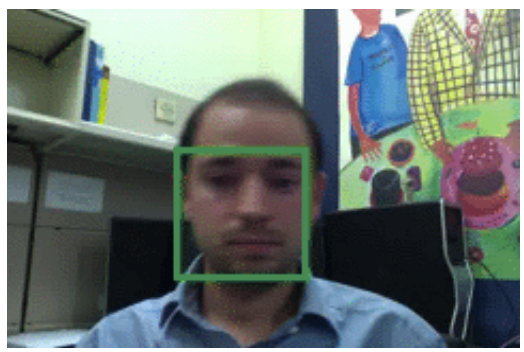
Slide credit: D. Hoiem

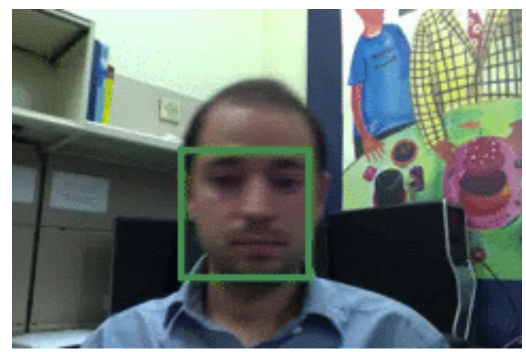
Tracking Issues

- Initialization
- Getting observation and dynamics models
- Combining prediction vs correction:
  - Dynamics too strong: ignores data
  - Observation too strong: tracking = detection



Too strong dynamics model





Too strong observation model

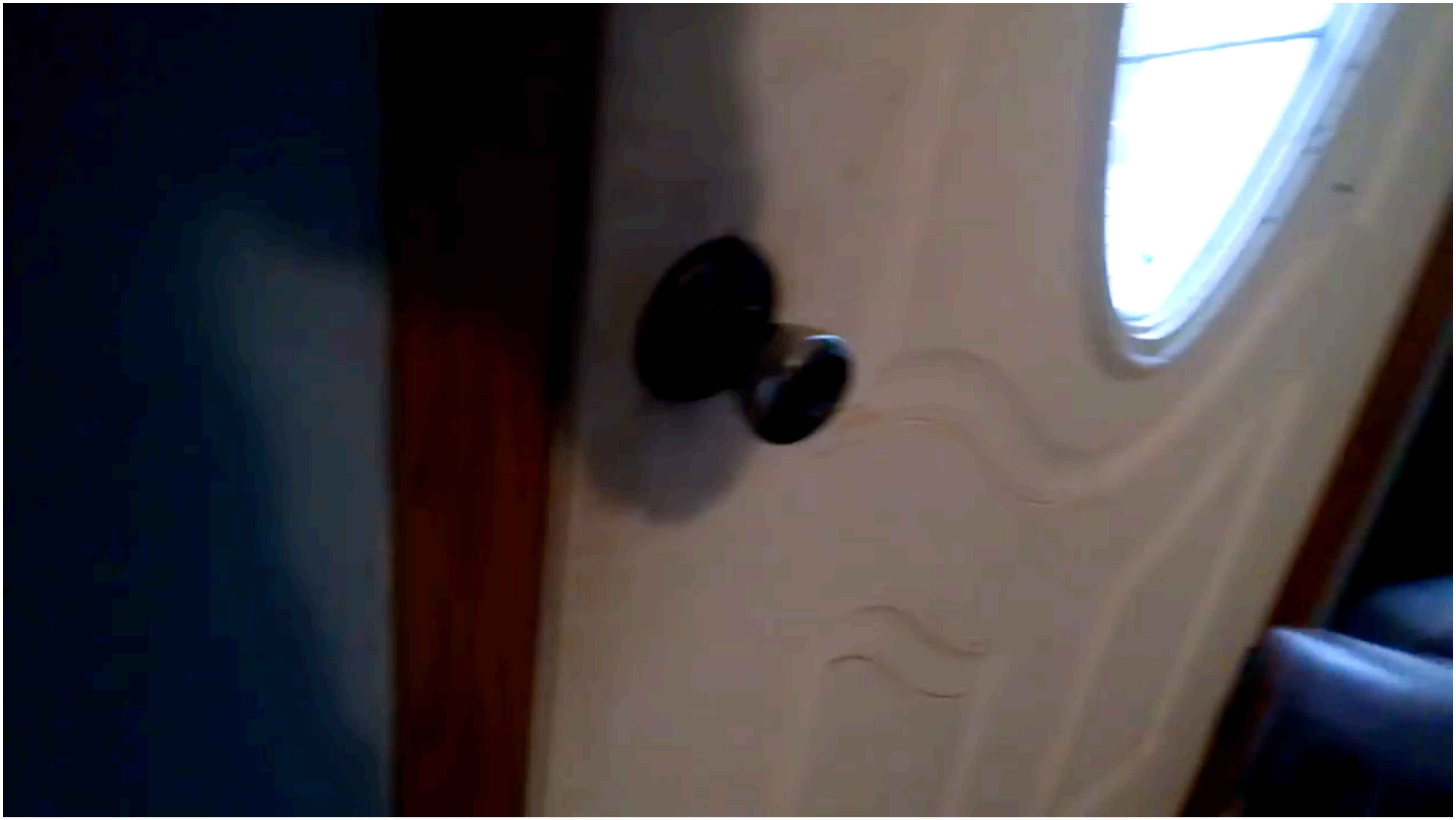
Slide credit: D. Hoiem

Tracking Issues

- Initialization
- Getting observation and dynamics models
- Combining prediction vs correction
- Data association:
  - Need to keep track of which object is which

Slide credit: D. Hoiem

Tracking Issues – Data Association

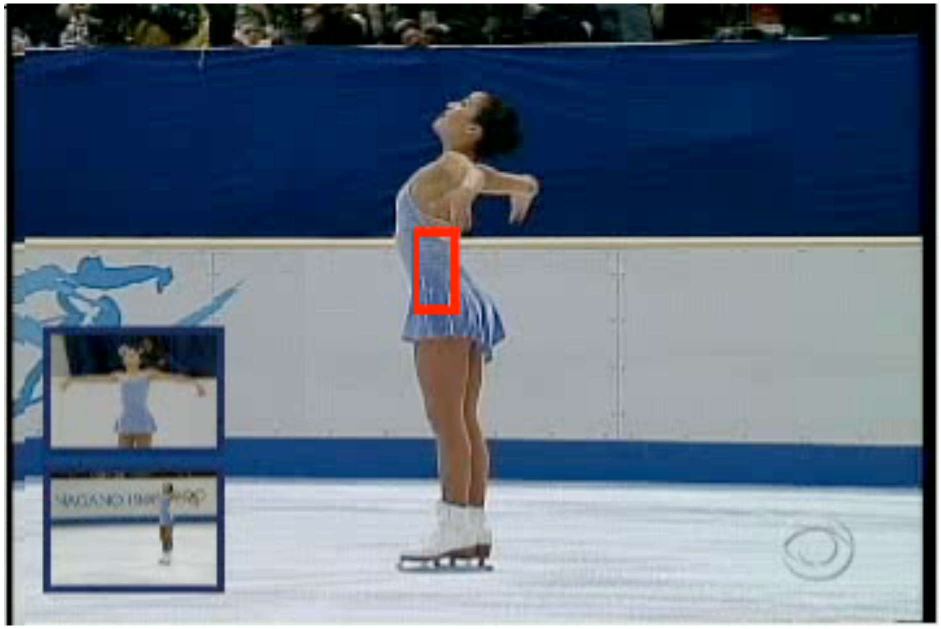


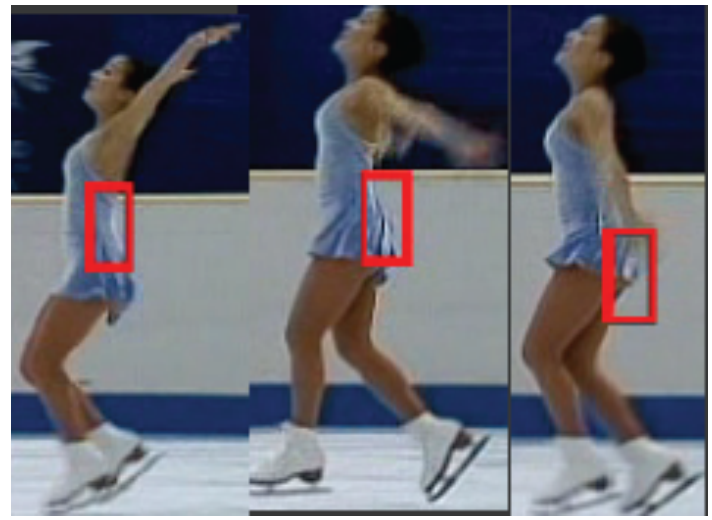
Tracking Issues

- Initialization
- Getting observation and dynamics models
- Combining prediction vs correction
- Data association
- Drift
  - Errors can accumulate over time

Slide credit: D. Hoiem

Drift





D. Ramanan, D. Forsyth, and A. Zisserman. [Tracking People by Learning their Appearance](). PAMI 2007.

Things to remember

- Tracking objects = detection + prediction
- Probabilistic framework
  - Predict next state
  - Update current state based on observation
- Two simple but effective methods
  - Kalman filters: Gaussian distribution
  - Particle filters: multimodal distribution

Slide credit: D. Hoiem

Next time: 3D